LDUIC - FIND THE LOAD UIC FOR N MACRO V05.03b Saturday 29-Jun-85 00:21 Page 6-3
Symbol table

```
U.CNT   000000          U.MLUN= 000050          U.TFRQ  000036          U.UTIL= 000036          U2.PRV= 000010
U.COTQ  000000          U.MUP   177772          U.TIXL  000060          U.VCB = 000034          U2.RMT= 020000
U.CTCB  000000          U.OWN   177776          U.TLPP  000053          U.VSER= 000120          U2.RO4= 100000
U.CTL   000004          U.RBNS= 000112          U.TMTI  000047          U.2MED= 000070          U2.SCS= 000004
U.CTYP  000052          U.RCTC= 000113          U.TRCK= 000100          U2.AT.= 000020          U2.SLV= 000200
U.CW1   000010          U.RCTS= 000110          U.TSTA  000026          U2.CRT= 002000          U2.VT5= 000002
U.CW2   000012          U.RED   000002          U.TST5  000054          U2.DH1= 100000          U2.7CH= 010000
U.CW3   000014          U.RED2  000034          U.TST6  000056          U2.DJ1= 040000          U3.OPA= 100000
U.CW4   000016          U.SCB   000020          U.TTAB  000050          U2.DZ1= 000100          U3.PAR= 040000
U.CYL = 000104          U.SHST= 000076          U.TTYP  000046          U2.ESC= 001000          U3.UPC= 020000
U.DCB   000000          U.SHUN= 000074          U.TUX   000024          U2.HFF= 010000          U4.CR = 000100
U.FCDE= 000042          U.SPC = 000036          U.UHVR= 000107          U2.HLD= 000040          V$$CTR= 001000
U.FNUM= 000040          U.STS   000005          U.UIC   000044          U2.LOG= 000400          X$$CBT= 000000
U.GRP = 000102          U.ST2   000007          U.UNF = 000052          U2.LWC= 000001          $CBOMG= ****** GX
U.KCSR= 000032          U.SUB = 000036          U.UNIT  000006          U2.L3S= 000004          $DIV  = ****** GX
U.KCS6= 000034          U.TCHP  000042          U.UNSZ= 000112          U2.L8S= 010000          $LDUIC  000000RG
U.LUIC  177774          U.TCVP  000043          U.UNTI= 000060          U2.NEC= 004000          $$$   = 000062
U.MEDI= 000040          U.TFLK  000040          U.USVR= 000106
```

```
. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        000170    001   (RW,I,LCL,REL,CON)
DATA    000020    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12204 Words  ( 48 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.44
SY:LDUIC.V2,[132,134]LDUIC/CR/-SP=SY:[  ]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]LDUIC
```


LDUIC CREATED BY MACRO ON 29-JUN-85 AT 00:21 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

```
     405 000442  000762                          BR      25$
     406 000444  026167  000010  000136' 40$:    CMP     10(R1),LENGTH   ; IS SYMBOL WITHIN TABLE LIMITS?
     407 000452  103757                          BLO     30$             ; IF LO, YES
     408 000454                                   EMSG$   2P              ; PRINT ERROR MESSAGE
     409 000462  000752                          BR      25$
     410 000464  005402                  50$:    NEG     R2              ; SET C-BIT IF ANY ERRORS OCCURRED
     411 000466                          60$:    RETURN
     412
     413                                 ;
     414                                 ; ERROR CONDITIONS
     415                                 ;
     416 000470  016000  000154'         101$:   MOV     ETAB-2(R0),R0   ; GET ERROR MESSAGE BLOCK ADDRESS
     417 000474                                  CALLR   @(R0)+          ; PRINT ERROR MESSAGE
```

LDUIC CREATED BY MACRO ON 29-JUN-85 AT 00:21 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
D.NAM	000004		6-113	6-114	
D.UCB	000002		6-107		
D.UCBL	000010		6-108		
D.UNIT	000006		6-110		
NTUIC	= ******	GX	6-95	6-97	
R$$11M	= 000000		5-61		
TKTCB	= ******	GX	6-102		
T.LDV	000044		6-103		
UICBF	000000	R	#5-74	6-100	6-132
U.DCB	000000		6-106		
$CBOMG	= ******	GX	6-119	6-124	6-128
$DIV	= ******	GX	6-109		
$LDUIC	000000	RG	#6-93		

NTL -

LDUIC CREATED BY MACRO ON 29-JUN-85 AT 00:21 PAGE 2

NTL

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  FSRSZ$,GCLDF$

                                ;***
                                ; GCL FILE PROCESSING STRUCTURES
                                ;***
000000                                  GCLDF$  1,1,,,2,CL.OPN,CL.ATT!CL.CMT!CL.EXT!CL.IND
000000                                  FSRSZ$  1

                                ;
                                ; GLOBAL GCL SYMBOLS
                                ;
                                        .GLOBL  CL.OPN
```

```
                                        .RAD50  "$CCBRT"
                                        .WORD   0
000170  000202' 000000          VXBLK2: .WORD   VXBLK3,0
000174  124473  006654                  .RAD50  "$CCBGT"
000200  000000                          .WORD   0
000202  000000  000000          VXBLK3: .WORD   0,0
000206  126344  035115                  .RAD50  "$ZTIME"
000212  000000                          .WORD   0


                                ; COMM EXEC SYMBOLS USED TO INITIALIZE VARIABLES
                                ;
000214  000226' 000000          IBLK1:  .WORD   IBLK2,0
000220  124520  023547                  .RAD50  "$CXFLG"
000224  000000                          .WORD   0
000226  000240' 000000          IBLK2:  .WORD   IBLK3,0
000232  125721  074375                  .RAD50  "$SYSNM"
000236  000000                          .WORD   0
000240  000000  000000          IBLK3:  .WORD   0,0
000244  125404  102253                  .RAD50  "$NTUIC"
000250  000000                          .WORD   0


                                ; CETAB SYMBOLS TO INITIALIZE THE COMM EXEC
                                ;
000252  000264' 000000          .PDVTB: .WORD   .SLTMB,0
000256  125504  106242                  .RAD50  "$PDVTB"
000262  000000                          .WORD   0
000264  000276' 000000          .SLTMB: .WORD   .LLCTB,0
000270  125704  077412                  .RAD50  "$SLTMB"
000274  000000                          .WORD   0
000276  000310' 000000          .LLCTB: .WORD   SLTTOT,0
000302  125254  012742                  .RAD50  "$LLCTB"
000306  000000                          .WORD   0
000310  000322' 000000          SLTTOT: .WORD   PDVTOT,0
000314  074264  077554                  .RAD50  "SLTTOT"
000320  000000                          .WORD   0
000322  000000  000000          PDVTOT: .WORD   0,0
000326  062266  077554                  .RAD50  "PDVTOT"
000332  000000                          .WORD   0


                                ; SYMBOL TABLE ERRORS
                                ;
000334  000000G               ETAB1:  .WORD   $ERR44          ; OPEN FAILURE
000336  000000G                       .WORD   $ERR45          ; READ FAILURE
000340  000000G                       .WORD   $ERR46          ; READ OVERFLOW
000342  000000G                       .WORD   $ERR47          ; BAD STB FORMAT


                                ; PROCESS I/O ERRORS
                                ;
000344  000000G               ETAB2:  .WORD   $ERR51          ; OPEN FAILURE
000346  000101                         .WORD   101             ; GET ATTRIBUTES FAILED
000350  000000G                       .WORD   $ERR53          ; FILE NOT CONTIGUOUS
000352  000000G                       .WORD   $ERR54          ; LABEL BLOCK READ FAILURE
000354  000000G                       .WORD   $ERR55          ; INVALID IMAGE (HEADER, OVERLAID, ETC.)
000356  000000G                       .WORD   $ERR56          ; PROCESS NOT BUILD FOR APR5
```

```
                                        .SBTTL  CCBS - ALLOCATE AND LINK CCB'S

                                ;+
                                ; CCBS - ALLOCATE AND LINK CCB'S
                                ;
                                ; INPUTS:
                                ;
                                ;       $CCBNM  - NUMBER OF CCBS TO ALLOCATE
                                ;       $CCBSZ  - SIZE OF EACH CCB IN BYTES
                                ;
                                ; OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ;-

001736                          CCBS:
001736  017702  000000G         10$:    MOV     @CCBNM,R2               ; NUMBER OF CCBS TO ALLOCATE
001742  017701  000000G                 MOV     @CCBSZ,R1               ; NUMBER OF BYTES/CCB

                                        .IF DF  R$$MPL
                                        TSTB    @NCPU                   ; MULTIPROCESSOR SYSTEM?
                                        BEQ     20$                     ; BR IF NO
                                        ADD     #2,R1                   ; ALLOCATE TWO MORE BYTES
                                20$:    .ENDC

001746                                  CALL    $ALLI6                  ; ATTEMPT ALLOCATION FROM SCOM
001752  103412                         BCS     101$                    ; RESOURCE ERROR

                                        .IF DF  R$$MPL
                                        TSTB    @NCPU                   ; MULTIPROCESSOR SYSTEM?
                                        BEQ     30$                     ; BR IF NO
                                        TST     (R0)+                   ; POINT PAST NEGATIVE OFFSET
                                30$:    .ENDC

001754  017710  000000G               MOV     @CCBLH,(R0)             ; ADD CCB
001760  010077  000000G               MOV     R0,@CCBLH               ;  TO QUEUE
001764  112760  000002  000003        MOVB    #CB.CCB,C.BID(R0)       ; BUFFER ID
001772  005302                         DEC     R2                      ; LOOP TILL
001774  001362                         BNE     10$                     ;  DONE.
001776                                  RETURN

                                ;
                                ; ERROR CONDITION
                                ;
002000  017700  000000G         101$:   MOV     @CCBLH,R0               ; GET FIRST CCB ADDRESS
002004  001407                         BEQ     110$                    ; IF ZERO, DONE
002006  011077  000000G                 MOV     (R0),@CCBLH             ; SET NEW FIRST IN LIST
002012  017701  000000G                 MOV     @CCBSZ,R1               ; GET CCB SIZE

                                        .IF DF  R$$MPL
                                        TSTB    @NCPU                   ; MULTIPROCESSOR SYSTEM?
                                        BEQ     105$                    ; BR IF NO
                                        TST     -(R0)                   ; TWO EXTRA BYTES TO ALLOCATE (NEG OFFSET)
                                        TST     (R1)+                   ; UPDATE CCB SIZE
                                105$:   .ENDC
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
PR6	= 000300		#5-68								
PR7	= 000340		#5-68								
PS	= 177776		#5-68								
PUMR	= ******	GX	13-778	13-779							
P.BLKS	= ******	GX	7-331	13-735							
P.F	= ******	GX	7-330	13-734							
QE	= ******	GX	13-780								
QSTRT	= ******	GX	13-781								
RDBCT	= ******	GX	13-728								
RDBLH	= ******	GX	15-895	15-902	15-904						
RDBNM	= ******	GX	15-884								
RDBS	002032	R	13-796	#15-883							
RDBSZ	= ******	GX	15-885								
RDBTH	= ******	GX	13-731								
R$$EIS	= ******		7-333	13-755	17-985						
R$$MPL	= ******		5-68	5-68	5-68	6-126	6-135	6-169	7-306	8-404	8-428
			8-452	10-556	11-652	12-678	13-772	14-826	14-835	14-856	
R$$11D	= ******		5-68	7-333	13-755	17-985					
SDBCT	= ******	GX	13-729								
SDBLH	= ******	GX	16-942	16-948	16-950						
SDBNM	= ******	GX	16-930								
SDBS	002140	R	13-797	#16-929							
SDBSZ	= ******	GX	16-932								
SF.ACT	= 000200		#5-63								
SF.ENA	= 000100		#5-63								
SF.LPB	= 000004		#5-63								
SF.MFL	= 000040		#5-63								
SF.PAC	= 000020		#5-63								
SF.REA	= 000010		#5-63								
SF.SER	= 000001		#5-63								
SF.SVC	= 000002		#5-63								
SF.UNL	= 000040		#5-63								
SLTMA	= ******	GX	11-604	11-629	11-636	18-1027					
SLTNM	= ******	GX	11-606	11-637	11-642						
SLTTOT	000310	R	6-243	#6-246	11-606						
STBERR	000544	R	8-412	8-419	8-426	#8-483					
STBFXT	000020	R	#6-148	8-401	8-415	8-422					
SWR	= 177570		#5-68								
SYSNM	= ******	GX	8-448								
S$CEX	000000	RG	#7-303								
S.COST	000001		#5-63								
S.FLG	000000		#5-63								
S.LEN	000004		#5-63								
S.NMST	000002		#5-63								
S.OWNR	000003		#5-63								
TKPS	= ******	GX	11-598								
TKTOO	= ******	GX	11-601								
TPS	= 177564		#5-68								
T$KERR	000316	R	7-324	7-351	#7-377						
T$FLAG	000044		#5-62								
T$LIF	000013		#5-62								
T$LIFL	000013		#5-62								

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 5

```
                                        .SBTTL  $NLCFG - PROCESS CONFIGURATION FILE

                                ;+
                                ; $NLCFG - PROCESS CONFIG FILE FOR MISSING PARAMETERS
                                ;
                                ; THIS ROUTINE IS CALLED TO SCAN THE CONFIG FILE FOR "SET PROCESS" OF AN LLC
                                ; (LOOKING FOR LOGICAL LINK COUNT AND RSX11M DEVICE NAME) AND FOR ALL "SET LINE"
                                ; CASES (LOOKING FOR DEVICE HARDWARE/SOFTWARE CHARACTERISTICS).
                                ;
                                ; INPUTS:
                                ;       $MISS=MISSING PARAMETERS MASK
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000000                          $NLCFG::
000000  010667  000354'                MOV     SP,SPSAV        ; SAVE SP FOR ERROR EXIT

                                .IF DF  R$$MPL
                                CLR     URMFL           ; CLEAR THE UNIBUS RUN MASK FLAG
                                .ENDC

000004  005067  000136'                CLR     CFGFLG          ; CLEAR FLAGS BYTE
000010                                 CALL    OPEN            ; OPEN THE CONFIG FILE
000014  012700  000012'                MOV     #NAME,R0        ; CONVERT DDM/LLC NAME TO ASCII
000020  016701  000000G                MOV     $NAME,R1        ; AND STORE IN "NAME"
000024                                 CALL    $C5TA           ;       ...
000030  032767  000000G 000000G        BIT     #MS.CIR,$MISS   ; ARE CIRCUIT PARAMETERS MISSING?
000036  001457                         BEQ     20$             ; BR IF NO
000040  005067  000140'                CLR     SCNT            ; CLEAR LOCAL COUNTERS
000044  005067  000142'                CLR     UCNT            ;       ...
000050  016700  000000G                MOV     $SLTA,R0        ; GET SLT ADDRESS
000054  116067  000012  000015'        MOVB    L.CTL(R0),CTL   ; COPY CONTROLLER NUMBER
000062  116067  000013  000016'        MOVB    L.UNT(R0),UNT   ; AND UNIT NUMBER
000070                          2$:    CALL    READ            ; READ NEXT RECORD
000074  103002                         BCC     5$              ; BR IF OK
000076  000167  000610                JMP     111$            ; IF CS, EOF
000102                          5$:    CALL    $QSLT           ; IS IT SLT$DF?
000106  103770                         BCS     2$              ; BR IF NO - KEEP LOOKING
000110                          7$:    CALL    READ            ; GET NEXT RECORD
000114  103002                         BCC     8$              ; BR IF OK
000116  000167  000612                JMP     141$            ; IF CS, EOF
000122                          8$:    CALL    $QSTA           ; IS IT STA$DF?
000126  103370                         BCC     7$              ; IF CC, YES - LOOK FOR MORE OF THEM
000130                          10$:   CALL    READ            ; READ NEXT RECORD
000134  103002                         BCC     15$             ; BR IF OK
000136  000167  000622                JMP     181$            ; BR IF EOF
000142                          15$:   CALL    $QSVC           ; IS IT SVC$DF?
000146  103011                         BCC     17$             ; BR IF YES
000150                                 CALL    $QX3P           ; IS IT X3P$DF?
000154  103365                         BCC     10$             ; BR IF YES
000156                                 CALL    $QDTE           ; IS IT DTE$DF?
000162  103362                         BCC     10$             ; BR IF YES
000164                                 CALL    $QPVC           ; IS IT PVC$DF?
000170  103757                         BCS     10$             ; BR IF NO
000172  000167  000476          17$:   JMP     190$            ; ELSE FINISH UP
```

```
                                        .SBTTL  PLT$DF STATE TABLE
                                ;
                                ; NUMBER OF CHANNELS TO ALLOCATE FOR AN NI INTERFACE
                                ;
001402                          STATE$  PLTDF
001402                           TRAN$  %PLT$DF%,,,1,SYNERR
001402                          STATE$
001402                           TRAN$  $NUMBR,$EXIT,PL.CHN

001402                          STATE$
```

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
MS.MTP	= ******	GX	9-318								
MS.MUX	= ******	GX	9-302								
M$$MGE	= 000000		20-717	20-728							
NAME	000012	RG	#7-109	9-235	15-581	20-698					
OPEN	000772	R	9-234	#10-420							
PDVNM	= ******	GX	25-834								
PDVTA	= ******	GX	25-832	25-847							
PLTDF	000244	R	12-479								
PL.CHN	001566	R	#22-756								
P.EXT	001506	R	#20-729								
P.INC	001466	RG	#20-718								
QADD	001134	R	9-317	#12-482							
QLLC	001142	R	9-355	#12-485							
QPLT	001126	R	9-315	#12-479							
READ	001034	R	9-245	9-250	9-255	9-279	9-289	9-296	9-304	9-311	9-325
			9-334	9-344	9-353	9-358	9-369	9-379	#11-446		
REP8A	001332	RG	#15-578	15-580							
RTSPC	001310	RG	8-172	8-173	8-174	8-175	8-176	8-177	8-178	8-179	8-180
			8-181	8-182	8-183	8-184	8-185	8-186	8-187	8-188	8-189
			8-190	8-191	8-192	#13-543					
R$$MPL	= ******		7-144	9-229							
SCNT	000140	RG	#7-127	*9-240	*9-274						
SF.ACT	= 000200		#5-64								
SF.ENA	= 000100		#5-64								
SF.LPB	= 000004		#5-64								
SF.MFL	= 000040		#5-64								
SF.PAC	= 000020		#5-64								
SF.REA	= 000010		#5-64								
SF.SER	= 000001		#5-64								
SF.SVC	= 000002		#5-64								
SF.UNL	= 000040		#5-64								
SPSAV	000354	R	#7-139	*9-227	13-541						
SYNERR	000356	RG	#7-151	*12-491	*12-494	*20-711					
S$$BAS	= ******		8-170	8-170	8-171	8-171	8-172	8-172	8-173	8-173	8-174
			8-174	8-175	8-175	8-176	8-176	8-177	8-177	8-178	8-178
			8-179	8-179	8-180	8-180	8-181	8-181	8-182	8-182	8-183
			8-183	8-184	8-184	8-185	8-185	8-186	8-186	8-187	8-187
			8-188	8-188	8-189	8-189	8-190	8-190	8-191	8-191	8-192
			8-192								
S.COST	000001		#5-64								
S.FLG	000000		#5-64								
S.LEN	000004		#5-64								
S.NAME	001402	RG	#20-698								
S.NXST	000002		#5-64								
S.OWNR	000003		#5-64								
UCNT	000142	RG	#7-128	*9-241	*9-275	9-300	*9-333	9-338			
UNT	000016	RG	#7-111	*9-244	*9-278						
WNDSMX	= 000177		#6-88	27-898							
Z$.PSE	= ******	GX	9-321								
Z.FLG	= ******	GX	9-321								
Z.NAM	= ******	GX	25-839								
$ALPHA	= 000022		#16-596								

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 4

```
                                                BCS     42$             ; IF CS, END OF FILE
                                                CLR     MRKFL           ; CLEAR LOCAL FLAG
                                                CALL    QMFL            ; LOOK FOR MARKED-FOR-LOAD/ENABLE
                                                CALL    $QDEC           ; LOOK FOR DEC$DF MACRO
                                                BR      40$             ; LOOP UNTIL END OF FILE
                                        42$:
                                                .IF     DF, R$$11M
                                                MOV     $RSXNM,R0       ; GET ADDRESS OF SYSTEM NAME
                                                BEQ     75$             ; BR IF NOT DEFINED
                                                MOV     @DECPT,R1       ; STORE DECNET HOME BLOCK OFFSET
                                                CLR     D$NBRA(R1)      ; ASSUME ENDNODE
                                                TST     D$NN(R1)        ; ENDNODE ?
                                                BEQ     420$            ; IF YES - BRANCH
                                                MOV     NBRA,D$NBRA(R1) ; IF NO - SET UP BROADCAST ROUTER ADJ. TOTAL
                                        420$:   ADD     #D$LNAM,R1      ; POINT TO NODE NAME
                                                TSTB    (R1)            ; WAS A NODE NAME SPECIFIED?
                                                BEQ     75$             ; BR IF NO
                                                CMPB    (R1),#40        ; A NON-BLANK NODE NAME?
                                                BNE     400$            ; IF YES - BRANCH

                                                SWSTK$  75$             ; ENTER SYSTEM STATE
                                                MOV     @DECPT,R0       ; PSI ONLY SYSTEM
                                                MOV     #D$END,R1       ; GET HOME BLOCK SIZE
                                                CLR     @DECPT          ; INIT HOME BLOCK POINTER
                                                CALL    @DEACB          ; DEALLOCATE HOME BLOCK
                                                RETURN

                                        400$:   MOV     #6,R2           ; NODE NAME IS 6 CHARACTERS LONG
                                                MOV     R0,-(SP)        ; SAVE STARTING ADDRESS OF SYSTEM NAME
                                                MOV     R1,-(SP)        ; SAVE NODE NAME ADDRESS
                                                MOV     R2,-(SP)        ; SAVE NODE NAME LENGTH
                                        421$:   CMPB    (R0)+,(R1)+     ; DO NAMES MATCH?
                                                BNE     422$            ; BR IF NO
                                                DEC     R2              ; MORE TO CHECK?
                                                BGT     421$            ; BR IF YES
                                                CMP     (SP)+,(SP)+     ; CLEAN UP STACK
                                                TST     (SP)+           ; ...
                                                BR      75$             ; ELSE NODE NAMES MATCH
                                        422$:   MOV     (SP)+,R2        ; SAVE ADDRESS OF SYSTEM NAME
                                                MOV     (SP)+,R1        ; GET NODE NAME ADDRESS
                                                MOV     (SP)+,R0        ; GET NODE NAME LENGTH

                                                SWSTK$  44$             ;; ENTER SYSTEM STATE
                                        43$:    MOVB    (R1)+,(R0)+     ;; UPDATE SYSTEM NAME
                                                DEC     R2              ;; MORE TO COPY?
                                                BGT     43$             ;; BR IF YES
                                                RETURN                  ;; RETURN TO USER STATE

                                        44$:    EMSG$   17              ; TELL USER NAME WAS CHANGED
                                                .ENDC
                                                BR      75$             ; FINISH UP
                                        ;
                                        ; NTINIT PHASE 1 SCAN
                                        ;
                                        45$:    INC     .NOMAP          ; DEFAULT SIZE OF NON-UMR-MAPPED AREA
                                                INC     .FUDGE          ; AND FUDGE FACTOR SIZE
                                        50$:    CALL    READ            ; READ NEXT RECORD
```

```
    827 001252                          TRAN$  <',>,E
    828 001252                          TRAN$  $STRNG,,SERDV,SF.DEV,SERFLG
    829
    830 001252                        STATE$
    831 001252                          TRAN$  $EOS,$EXIT,SEREND
    832 001252                          TRAN$  <',>
```

```
                                        .SBTTL  TPARS STATE TABLES - REM$DF
                                ;
                                ; REMOTE DEFINITIONS (REM$DF)
                                ;
1243 001252                             STATE$  REMDF
1244 001252                              TRAN$   LBRA
1245
1246 001252                             STATE$                          ; REMOTE NODE NAME
1247 001252                              TRAN$   $STRNG,,R.NAME
1248
1249 001252                             STATE$
1250 001252                              TRAN$   RBRA
1251
1252 001252                             STATE$
1253 001252                              TRAN$   <',>
1254
1255 001252                             STATE$
1256 001252                              TRAN$  $NUMBR,,R.ARE    ; REMOTE NODE AREA
1257
1258 001252                             STATE$
1259 001252                              TRAN$  <',>
1260
1261 001252                             STATE$                          ; REMOTE NODE ADDRESS
1262 001252                              TRAN$   $NUMBR,$EXIT,R.NUM
1263
```

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 7

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
$RAD50	= 000016		#17-701		
$REP.N	= ******	GX	7-107	7-108	
$RONLY	= ******		17-701	17-701	17-701
$RSXNM	= ******	GX	8-332		
$SAVAL	= ******	GX	12-538		
$STRNG	= 000004		#17-701		
$SUBXP	= 000010		#17-701		
$TIOUT	= ******	GX	12-563	13-592	
$$$FLG	= 177777		#17-701		
$$$KEY	= 177777		#17-701		
.BRAD	002464	RC	#7-258		
.FUDGE	= ******	GX	*8-380		
.NOMAP	= ******	GX	*8-379		
.RDBSZ	= ******	GX	15-651		
.TPARS	= ******	GX	11-505		

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 8

```
                                        .SBTTL  OPEN - OPEN THE CONFIGURATION FILE

                                ;+
                                ; OPEN - OPEN THE CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000230  012700  000000G         OPEN:   MOV     #$CFUIC,R0      ; SETUP ASCIZ FILE SPEC
000234  012701  000000'                 MOV     #CFGNAM,R1      ; FOR [130,10]CETAB.MAC
000240  012702  000006'                 MOV     #CFGEXT,R2      ;       ...
000244                                  CALL    $AZFS           ;       ...
000250                                  CALL    $CLQUE          ; OPEN THE CONFIG FILE
000254  105777  000000G                 TSTB    @$CLIOS         ; SUCCESS? (CLEARS C-BIT)
000260  100401                          BMI     101$            ; IF MI, NO
000262                                  RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000264                          101$:   EMSG$R  1P              ; OPEN FAILURE
```

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ERRMSG	000012 R	#7-101 12-407
FL.NTI	= ****** GX	9-246
FMT10	001534 RG	8-200 8-201 8-202 8-203 8-206 8-207 8-208 #8-224
FMT8	001514 RG	8-204 8-205 8-209 8-210 8-211 8-212 8-213 8-214 8-215 #8-223
FM.10	= 000000	#8-200 #8-201 #8-202 #8-203 #8-206 #8-207 #8-208
FM.8	= 000000	#8-204 #8-205 #8-209 #8-210 #8-211 #8-212 #8-213 #8-214 #8-215
FNDPDV	001012 RG	#17-574
FNLIN	000500 RG	8-214 #13-433
HSHADD	000376 RG	#7-140
HSHSZ	000374 RG	#7-139
IE.EOF	= ****** GX	11-361
IE.RBG	= ****** GX	11-364
I$$AS	= ******	15-524
LBRA	= 000074	#6-71
LF.ACT	= 100000	#5-64
LF.BRO	= 000400	#5-64
LF.BWT	= 000007	#5-64
LF.ENA	= 002000	#5-64
LF.LPB	= 001000	#5-64
LF.MDC	= 000100	#5-64
LF.MFL	= 004000	#5-64
LF.MTP	= 000020	#5-64
LF.PAC	= 000200	#5-64
LF.RDY	= 040000	#5-64
LF.REA	= 010000	#5-64
LF.SER	= 000040	#5-64
LF.TIM	= 000010	#5-64
LF.UNL	= 020000	#5-64
LF.X2P	= 000000	#5-64
LINCTL	000350 RG	#7-117 *14-455
LINNAM	000346 RG	#7-116 *14-448
LINUNT	000351 RG	#7-118 *14-462
LNKEND	000760 RG	#16-548 16-554
LN.CLO	= 030000	#5-64
LN.DUM	= 000005	#5-64
LN.LOA	= 000004	#5-64
LN.LOO	= 000003	#5-64
LN.OAU	= 000003	#5-64
LN.OFF	= 000001	#5-64
LN.ON	= 000000	#5-64
LN.OOP	= 000004	#5-64
LN.OPE	= 000001	#5-64
LN.REF	= 000002	#5-64
LN.SER	= 000002	#5-64
LN.STA	= 000017	#5-64
LN.SUB	= 000360	#5-64
LN.TRI	= 000006	#5-64
LO.CON	000006	#5-63
LO.CSR	000006	#5-63
LO.CTL	000014	#5-63
LO.LIN	000004	#5-63

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 3

```
**FILE**ID**NTLDEV
```

```
NN       NN   TTTTTTTTTT   LL           DDDDDDDD     EEEEEEEEEE   VV       VV
NN       NN   TTTTTTTTTT   LL           DDDDDDDD     EEEEEEEEEE   VV       VV
NN       NN       TT       LL           DD       DD  EE           VV       VV
NN       NN       TT       LL           DD       DD  EE           VV       VV
NNNN     NN       TT       LL           DD       DD  EE           VV       VV
NNNN     NN       TT       LL           DD       DD  EE           VV       VV
NN  NN   NN       TT       LL           DD       DD  EEEEEEEE     VV       VV
NN  NN   NN       TT       LL           DD       DD  EEEEEEEE     VV       VV
NN     NNNN       TT       LL           DD       DD  EE           VV       VV
NN     NNNN       TT       LL           DD       DD  EE           VV       VV
NN       NN       TT       LL           DD       DD  EE             VV  VV         ....
NN       NN       TT       LL           DD       DD  EE             VV  VV         ....
NN       NN       TT       LLLLLLLLLL   DDDDDDDD     EEEEEEEEEE       VV           ....
NN       NN       TT       LLLLLLLLLL   DDDDDDDD     EEEEEEEEEE       VV           ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .SBTTL  DVLBL - OPEN FILE AND READ LABEL BLOCK

                                ;+
                                ; DVLBL - OPEN IMAGE FILE AND READ LABEL BLOCK
                                ;
                                ; THIS ROUTINE IS CALLED TO OPEN THE DATA STRUCTURE FILE AND TO READ THE LABEL
                                ; BLOCK. THE SYMBOL VALUES FROM THE STB FILE ARE ADJUSTED BY SUBTRACTING THE
                                ; BASE ADDRESS OF THE DATA STRUCTURE IMAGE FILE.
                                ;
                                ; INPUTS:
                                ;       DTNAM=FILE NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=LABEL BLOCK BUFFER ADDRESS
                                ;-
000476  005000                  DVLBL:  CLR     R0              ; SETUP ASCIZ FILE SPEC
000500  005001                          CLR     R1              ; FOR [131,54]XXXTAB.TSK
000502  005002                          CLR     R2              ;       ...
000504                                  CALL    $AZFS           ;       ...
000510                                  CALL    $CLOPE          ; OPEN THE IMAGE FILE
000514  105777  000000G                 TSTB    @$CLIOS         ; SUCCESS?
000520  100427                          BMI     101$            ; IF MI, NO
000522  052010                          BIS     (R0)+,(R0)      ; IS FILE CONTIGUOUS?
000524  001430                          BEQ     121$            ; IF EQ, NO
000526  012700  000000'                 MOV     #DSKBUF,R0      ; GET ADDRESS OF DISK BLOCK SIZED BUFFER
000532                                  CALL    $RLBL           ; READ LABEL BLOCK
000536  103426                          BCS     131$            ; IF CS, FAILURE
000540  032760  000000G 000000G         BIT     #IS$NHD,L$BFLG(R0) ; DOES IMAGE HAVE A HEADER?
000546  001430                          BEQ     141$            ; IF EQ, YES - NOT ALLOWED
000550  026060  000000G 000000G         CMP     L$BLDZ(R0),L$BMXZ(R0) ; IS THE IMAGE OVERLAID?
000556  001024                          BNE     141$            ; IF NE, YES - NOT ALLOWED
000560  016067  000000G 000170'         MOV     L$BSA(R0),BASE  ; SAVE BASE ADDRESS FOR LATER
000566  166767  000170' 000136'         SUB     BASE,LENGTH     ; ADJUST DEVICE TABLES SIZE
000574  000241                          CLC                     ; INDICATE SUCCESS
000576                                  RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000600                          101$:   EMSG$R  2F              ; OPEN FAILURE
000606                          121$:   EMSG$R  2H              ; FILE NOT CONTIGUOUS
000614  116777  000000G 000000G 131$:   MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
000622                                  EMSG$R  2J              ; LABEL BLOCK READ FAILURE
000630                          141$:   EMSG$R  2K              ; INVALID BINARY IMAGE (HEADER OR OVERLAID)
```

LDUIC CREATED BY MACRO ON 29-JUN-85 AT 00:21 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES			
CALL	6-109	6-119	6-124	6-128
DCBDF$	#5-62	5-66		
RETURN	6-134			
TCBDF$	#5-62	5-64		
UCBDF$	#5-62	5-65		

NTL

FILEID**MAPX

NTL
Symb

```
     72          000001                          .END
```


Symbol table

```
000360  000000G                         .WORD   $ERR57          ; PROCESS EXCEEDS 4K WORDS
000362  000000G                         .WORD   $ERR58          ; PROCESS EXTENSION EXCEEDS 4K WORDS
000364  000000G                         .WORD   $ERR59          ; PARTITION NOT IN SYSTEM
000366  000000G                         .WORD   $ERR60          ; BASE ADDRESS MISMATCH
000370  000000G                         .WORD   $ERR61          ; PARTITION TOO SMALL
000372  000000G                         .WORD   $ERR62          ; NOT COMMON PARTITION
000374  000000G                         .WORD   $ERR63          ; PARTITION BUSY
000376  000000G                         .WORD   $ERR64          ; CEX PARTITION NOT IN EXEC SPACE
000400  000000G                         .WORD   $ERR65          ; SUB-PCB ALLOCATION FAILURE
000402  000000G                         .WORD   $ERR66          ; MAIN PARTITION TOO FRAGMENTED
000404  000000G                         .WORD   $ERR67          ; PROCESS READ FAILURE
000406  000000G                         .WORD   $ERR68          ; CETAB ALLOCATION FAILURE

000000                                  .PSECT
```

```
    862
    863 002016                              CALL    $DEA16          ; DE-ALLOCATE CCB
    864 002022  000766                      BR      101$            ; RETURN THEM ALL
    865 002024                      110$:   EMSG$R  XQ              ; CCB ALLOCATION FAILURE
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES				
T$LIFO	000013		#5-62				
T$LIFS	000013		#5-62				
T$LIN	000000		#5-62				
T$LIPS	000006		#5-62				
T$LLD	000012		#5-62				
T$LLDC	000045		#5-62				
T$LLDL	000012		#5-62				
T$LLDO	000012		#5-62				
T$LLDS	000012		#5-62				
T$LLEN	000046		#5-62	18-1067			
T$LOPR	000002		#5-62				
T$LTCL	000024		#5-62				
T$LTIM	000026		#5-62				
T$LTPR	000014		#5-62				
T$LTPS	000020		#5-62				
T$NAPL	000004		#5-62				
T$NFE	000000		#5-62				
T$NLEN	000010		#5-62				
T$NNUL	000002		#5-62				
T$NOPL	000006		#5-62				
T$NRNI	000042		#5-62	*18-1073			
T$NRPL	000005		#5-62				
T$NRUL	000007		#5-62				
T$NVR	000001		#5-62				
T$RPRI	000040		#5-62	*18-1074			
T$SVC	000034		#5-62				
T$T5	000030		#5-62	*18-1071			
T$T6	000032		#5-62	*18-1072			
UBMPR	= 170200		#5-68				
UISAR0	= 177640		#5-68				
UISAR1	= 177642		#5-68				
VXBLK1	000156	R	#6-212	10-567			
VXBLK2	000170	R	6-212	#6-215			
VXBLK3	000202	R	6-215	#6-218			
XAVL	= ******	GX	13-744	13-751			
XBIAS	= ******	GX	13-743	13-750			
XBLK1	000120	R	6-189	#6-198	8-470	10-562	
XBLK2	000132	R	6-198	#6-201			
XBLK3	000144	R	6-201	#6-204			
XPTPDV	000034	R	#6-164	*18-1017	18-1041	18-1053	
ZTIME	= ******	GX	7-359				
Z.DAT	= ******	GX	18-1058				
Z.MAP	= ******	GX	18-1342				
$ALL16	= ******	GX	14-852				
$ALL18	= ******	GX	13-790				
$ALTLC	002314	RG	#18-1009				
$AZFS	= ******	GX	7-310	7-349	8-402	8-416	8-423
$CLINS	= ******	GX	6-177				
$CTBIO	= ******	GX	7-350				
$CXUIC	= ******	GX	7-307	7-346	8-399	8-413	8-420
$DEA16	= ******	GX	14-863				
$DIV	= ******	GX	11-600				

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 6

```
                                                    ;
000176  032767  000000G 000000G 20$:    BIT     #MS.LLC,$MISS   ; ARE LLC PARAMETERS MISSING?
000204  001160                         BNE     130$            ; IF NE, YES
000206  032767  000000C 000000G         BIT     #<MS.INC!MS.EXT>,$MISS  ; ARE THE PROCESS EXTENDS MISSING?
                                                                ; (SPECIAL CASE)
                                                                ; NOTE - MS.INC/EXT ONLY BIT SET FOR A PROCESS,
                                                                ; IF PROCESS IS AN LLC HANDLED SEPARATELY.
000214  001145                         BNE     120$            ; IF NE, YES - LOOK FOR IT
000216  005067  000140'               CLR     SCNT            ; CLEAR LOCAL COUNTERS
000222  005067  000142'               CLR     UCNT            ;   ...
000226  016700  000000G               MOV     $SLTA,R0        ; GET SLT ADDRESS
000232  116067  000012  000015'       MOVB    L.CTL(R0),CTL   ; COPY CONTROLLER NUMBER
000240  116067  000013  000016'       MOVB    L.UNT(R0),UNT   ; AND UNIT NUMBER
000246                          30$:    CALL    READ            ; READ NEXT RECORD
000252  103002                         BCC     35$             ; BR IF OK
000254  000167  000432                 JMP     111$            ; IF CS, EOF
000260                          35$:    CALL    $QSLT           ; IS IT SLT$DF?
000264  103042                         BCC     70$             ; IF CC, YES
000265                                  CALL    $QDDM           ; IS IT DDM$DF?
000272  103765                         BCS     30$             ; IF CS, NO
                                    ;
                                    ; DDM$DF SEEN
                                    ;
000274                          40$:    CALL    READ            ; READ NEXT RECORD
000300  103002                         BCC     45$             ; BR IF SUCCESSFUL
000302  000167  000412                 JMP     121$            ; ELSE EOF
000306                          45$:    CALL    $QCNT           ; IS IT CNT$DF?
000312  103770                         BCS     40$             ; IF CS, NO - KEEP LOOKING
000314  005767  000000G               TST     $MISS           ; ANY PARAMETERS STILL MISSING?
000320  001462                         BEQ     100$            ; IF EQ, NO
000322                          50$:    CALL    READ            ; READ NEXT RECORD
000326  103577                         BCS     131$            ; IF CS, EOF
000330                                  CALL    $QUNT           ; IS IT UNT$DF?
000334  103563                         BCS     107$            ; IF CS, NO - ERROR
000336  026767  000000G 000142'       CMP     $$MUX,UCNT      ; HAVE ALL UNT$DF'S BEEN SEEN?
000344  001366                         BNE     50$             ; IF NE, NO
000346  042767  000000G 000000G       BIC     #MS.MUX,$MISS   ; INDICATE ALL MUX PARAMETERS SEEN
000354  001444                         BEQ     100$            ; IF EQ, ALL DONE
000356                          60$:    CALL    READ            ; READ NEXT RECORD
000362  103553                         BCS     111$            ; IF CS, EOF
000364                                  CALL    $QSLT           ; IS IT SLT$DF?
000370  103772                         BCS     60$             ; IF CS, NO
                                    ;
                                    ; SLT$DF SEEN
                                    ;
000372                          70$:    CALL    READ            ; GET NEXT RECORD
000376  103556                         BCS     14[illegible]   ; IF CS, EOF
000400                                  CALL    $Q[illegible]   ; IS IT STA$DF?
000404  103372                         BCC     70$             ; IF CC, YES - LOOK FOR MORE OF THEM
000406                                  CALL    QPLT            ; IS IT PLT$DF ?
000412  103002                         BCC     75$             ; IF YES - BRANCH
000414                                  CALL    QADD            ; IS IT ADD$DF?
000420  042767  000000G 000000G 75$:    BIC     #MS.MTP,$MISS   ; INDICATE ALL MTP PARAMETERS SEEN
000426  001417                         BEQ     100$            ; IF EQ, ALL DONE
000430  016701  000000G               MOV     $PDVA,R1        ; GET PDV ADDRESS
000434  032761  000000G 000000G       BIT     #ZF.PSE,Z.FLG(R1) ; IS IT A PSEUDO DDM?
000442  001011                         BNE     100$            ; BR IF YES
000444                          80$:    CALL    $QDDM           ; IS IT DDM$DF?
```

```
                                        .SBTTL COMMON ACTION ROUTINES
                                ;
                                ; TPARS ACTION ROUTINES
                                ;
                                ; DDM NAME (SLT$DF, DDM$DF, LLC$DF)
                                ;
001402                          L.NAME:
001402                          D.NAME::
001402 012700 000012'           S.NAME::MOV     #NAME,R0        ; POINT AT NAME WE'RE LOOKING FOR
001406 016701 000000G                   MOV     .PSTPT,R1       ; POINT AT NAME FROM CONFIG FILE
001412 016702 000000G                   MOV     .PSTCN,R2       ; GET CONFIG FILE NAME LENGTH
001416 022702 000003                    CMP     #3,R2           ; THREE CHARACTERS?
001422 001407                           BEQ     10$             ; IF EQ, YES
001424 022702 000002                    CMP     #2,R2           ; TWO CHARACTERS?
001430 001011                           BNE     20$             ; IF NE, NO - REJECT TRANSITION
001432 122760 000040 000002             CMPB    #' ,2(R0)       ; IS OUR THIRD CHARACTER A SPACE?
001440 001005                           BNE     20$             ; IF NE, NO - REJECT TRANSITION
001442 122021                   10$:    CMPB    (R0)+,(R1)+     ; SEE IF NAMES MATCH
001444 001003                           BNE     20$             ; FOR A LENGTH OF R2 CHARACTERS
001446                                  SOB     R2,10$          ;   ...
001452 000404                           BR      30$             ; SUCCESS!
001454                          20$:    REJ$                    ; REJECT TRANSITION
001464                          30$:    RETURN
                                ;
                                ; PROCESS LINE TABLE COUNT (LLC$DF, DLC$DF, DDM$DF)
                                ;
                                        .IF DF  M$$MGE
001466                          P.INC::
001466 032767 000000G 000000G           BIT     #MS.INC,$MISS   ; IS THE LINE TABLE COUNT MISSING ?
001474 001403                           BEQ     10$             ; IF EQ, NO - LEAVE
001476 016767 000000G 000000G           MOV     .PNUMB,$$INC    ; GET THE LINE TABLE COUNT
001504                          10$:    RETURN
                                        .ENDC
                                ;
                                ; LLC PROCESS EXTENSION
                                ;
                                        .IF DF  M$$MGE
001506                          P.EXT:
001506 016767 000000G 000000G           MOV     .PNUMB,$$EXT    ; GET THE PROCESS EXTENSION SIZE
001514 026727 000000G 000200            CMP     .PNUMB,#128.    ; IS PROCESS EXTENSION >= 4K ?
001522 103001                           BHIS    101$            ; IF HIS, YES
001524                                  RETURN
                                ;
001526                          101$:   EMSG$R  ZQ              ; PROCESS EXTENSION TOO BIG
                                        .ENDC
```

SYMBOL	VALUE		REFERENCES						
$ANY	= 000020		#16-596						
$AZFS	= ******	GX	10-423						
$BLANK	= 000006		#16-596						
$CEACX	= ******	GX	26-869						
$CFUIC	= ******	GX	10-420						
$CLBUF	= ******	GX	*11-446						
$CLDEQ	= ******	GX	9-392	13-519					
$CLIOS	= ******	GX	10-425	11-447					
$CLQUE	= ******	GX	10-424						
$CLSIZ	= ******	GX	*11-446						
$CTIM	= ******	GX	*24-803						
$C5TA	= ******	GX	9-237						
$DIGIT	= 000024		#16-596						
$DNUMB	= 000014		#16-596						
$EOS	= 000012		#16-596						
$ERRYP	001356	RG	#8-187						
$ERRYR	001400	RG	#8-188	24-805					
$ERRZP	001434	RG	#8-189						
$ERRZQ	001304	RG	#8-186	20-735					
$ERR1A	001062	R	#8-181	22-764					
$ERR1P	000370	R	#8-170	10-432					
$ERR1Q	000426	R	#8-171	11-461					
$ERR1R	000464	R	#8-172	11-462					
$ERR1S	000512	R	#8-173	9-399					
$ERR1T	000542	RG	#8-174	12-501					
$ERR1U	000570	R	#8-175	9-400					
$ERR1V	000626	R	#8-176	9-401					
$ERR1W	000664	R	#8-177	9-402					
$ERR1X	000722	R	#8-178	9-403					
$ERR1Y	000760	R	#8-179	9-327					
$ERR1Z	001024	R	#8-180	9-404					
$ERR2A	001124	RG	#8-182	21-749					
$ERR2C	001174	RG	#8-183						
$ERR2F	001230	RG	#8-184						
$ERR2I	001264	RG	#8-185						
$ERR3A	001466	R	#8-190	9-405					
$ERR3B	001524	R	#8-191	9-406					
$ERR3C	001562	R	#8-192	9-407					
$EXIT	= 000000		#16-596						
$FAIL	= 177777		#16-596						
$FEATR	= ******	GX	*9-357						
$FNOUT	= ******	GX	8-170	8-171	14-558				
$GCLEP	= ******	GX	11-446						
$GPRM	= ******		16-596						
$LAMDA	= 000000		#16-596						
$LINKX	= ******	GX	26-874						
$MISS	= ******	GX	9-238	9-267	9-269	9-294	*9-302	*9-318	20-719
$NAME	= ******	CX	9-236						
$NLCFG	000000	RG	#9-226						
$NUMBR	= 000002		#16-596						
$PDVA	= ******	GX	9-320						
$QCNT	= ******	GX	9-292						

```
    382 000344  103450                          BCS     161$            ; IF CS, EOF (BUF$DF MISSING)
    383 000346                                   CALL    QPAR            ; IS IT PAR$DF ?
    384 000352  103053                           BCC     181$            ; YES .. BUF$DF IS MISSING
    385 000354                                   CALL    $QBRA           ; CHECK FOR SLT WITH ROUTER ADJACENCY VALUE
    386 000360                                   CALL    $QXPT           ; IS IT XPT'S LLC$DF (IF YES - GET # OF CKTS)
    387 000364                                   CALL    $QMLMC          ; IS IT DEC$DF (IF YES - GET # OF LNKS / CTRS)
    388 000370                                   CALL    QROU            ; IS IT ROU$DF ?
    389 000374  103361                           BCC     50$             ; IF CC, YES .. OKAY
    390 000376                                   CALL    QBUF            ; IS IT BUF$DF ?
    391 000402  103756                           BCS     50$             ; IF CS, NO
    392 000404  042767  000000G 000000G          BIC     #MS.BUF,$MISS   ; INDICATE WE FOUND IT
    393 000412                          60$:     CALL    READ            ; READ NEXT RECORD
    394 000416  103426                           BCS     171$            ; IF CS, EOF (PAR$DF MISSING)
    395 000420                                   CALL    QROU            ; IS IT ROU$DF ?
    396 000424  103372                           BCC     60$             ; IF CC, YES .. OKAY
    397 000426                                   CALL    QPAR            ; IS IT PAR$DF ?
    398 000432  103767                           BCS     60$             ; IF CS, NO
    399 000434  042767  000000G 000000G          BIC     #MS.PAR,$MISS   ; INDICATE WE FOUND IT
    400 000442                          65$:     CALL    READ            ; READ NEXT RECORD
    401 000446  103403                           BCS     70$             ; IF CS, EOF
    402 000450                                   CALL    QROU            ; IS IT ROU$DF ?
    403 000454  103772                           BCS     65$             ; IF CS, NO - KEEP LOOKING
    404                                 ;
    405                                 ; ALL NEEDED PARAMETERS SEEN
    406                                 ;
    407 000456                          70$:     CALL    $CLDEQ          ; CLOSE CONFIG FILE
    408 000462  000241                  75$:     CLC                     ; INDICATE SUCCESS
    409 000464                                   RETURN
    410
    411                                 ;
    412                                 ; ERROR CONDITIONS
    413                                 ;
    414 000466                          161$:    EMSG$R  XG              ; BUF$DF MISSING (EOF)
    415 000474                          171$:    EMSG$R  XH              ; PAR$DF MISSING (EOF)
    416 000502                          181$:    EMSG$R  XI              ; BUF$DF MISSING
```



```
OPEN - OPEN THE CONFIG FILE
```

```
                                ;+
                                ; SER$DF STATE TABLE CONTINUED
                                ;-


001252                                  STATE$  E
001252                                   TRAN$  <',>,F
001252                                   TRAN$  $STRNG,,PAS,SF.PSS,SERFLG

001252                                  STATE$
001252                                   TRAN$  $EOS,$EXIT,SEREND
001252                                   TRAN$  <',>

001252                                  STATE$  F
001252                                   TRAN$  <',>,G
001252                                   TRAN$  $NUMBR,,DADDR,SF.DPA,SERFLG

001252                                  STATE$
001252                                   TRAN$  $EOS,$EXIT,SEREND
001252                                   TRAN$  <',>

001252                                  STATE$  G
001252                                   TRAN$  <',>,H
001252                                   TRAN$  $NUMBR,,DCT,SF.DPC,SERFLG

001252                                  STATE$
001252                                   TRAN$  $EOS,$EXIT,SEREND
001252                                   TRAN$  <',>

001252                                  STATE$  H
001252                                   TRAN$  <',>,I
001252                                   TRAN$  $STRNG,,HRDADD,SF.HAD,SERFLG


001252                                  STATE$
001252                                   TRAN$  $EOS,$EXIT,SEREND
001252                                   TRAN$  <',>


001252                                  STATE$  I
001252                                   TRAN$  %SF.PH3%,$EXIT,SEREND,SF.PH3,SERFLG

001252                                  STATE$
001252                                   TRAN$  !END



001252                                  STATE$  SERST
001252                                   TRAN$  <'.>,MULP
001252                                   TRAN$  <'->

001252                                  STATE$
001252                                   TRAN$  $DNUMB,ZSS,SERMUX

001252                                  STATE$  MULP
001252                                   TRAN$  $DNUMB,,SERMUL
```

```
                                        .SBTTL  TPARS STATE TABLES - EVT$DF
                                ;
                                ; EVENT FILTER DEFINITIONS (EVT$DF)
                                ;
                                        STATE$  EVTDF
                                         TRAN$   $NUMBR,,E.NUM

                                        STATE$
                                         TRAN$   <',>

                                        STATE$
                                         TRAN$   LBRA

                                        STATE$
                                         TRAN$   !EVTMSK

                                        STATE$
                                         TRAN$   <',>

                                        STATE$
                                         TRAN$   "CONSOLE",$EXIT,E.CON
                                         TRAN$   "FILE",$EXIT,E.FIL
                                         TRAN$   "MONITOR",$EXIT,E.MON

                                ;
                                ; EVTMSK SUB-EXPRESSION
                                ;
                                        STATE$  EVTMSK
                                         TRAN$   $NUMBR,,E.MSK

                                        STATE$
                                         TRAN$   RBRA,$EXIT
                                         TRAN$   <',>,EVTMSK
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-59									
CALL	8-300	8-324	8-327	8-328	8-345	8-349	8-367	8-373	8-381	8-383
	8-385	8-386	8-387	8-388	8-390	8-393	8-395	8-397	8-400	8-402
	8-407	9-434	9-435	10-459	11-505	12-538	12-543	12-550	12-563	12-566
	13-589	14-621	14-624	15-653						
CALLR	13-592									
DBGTP$	#17-701	#17-706	#17-707	#17-711	#17-715	#17-719	#17-723	#17-727	#17-731	#17-735
	#18-747	#18-748	#18-755	#18-759	#18-760	#18-761	#19-779	#19-780	#19-787	#19-791
	#19-795	#19-802	#19-806	#19-810	#19-820	#19-824	#19-828	#19-832	#20-841	#20-845
	#20-849	#20-853	#20-857	#20-861	#20-865	#20-870	#20-883	#21-908	#21-909	#21-912
	#21-919	#22-932	#22-935	#22-938	#22-941	#22-944	#22-947	#22-950	#22-953	#22-956
	#22-959	#23-971	#23-972	#23-973	#23-974	#23-975	#23-976	#23-977	#23-978	#23-979
	#23-980	#23-981	#23-982	#23-1015	#24-1028	#24-1029	#24-1030	#24-1031	#24-1036	#24-1041
	#25-1064	#25-1067	#25-1068	#25-1069	#25-1072	#25-1081	#25-1084	#25-1087	#25-1090	#25-1093
	#25-1096	#25-1099	#26-1118	#26-1121	#26-1122	#26-1123	#26-1124	#26-1125	#27-1145	#27-1167
	#30-1286	#30-1287	#30-1297							
DHBDF$	#5-60	5-70								
EMSG$	#5-63	8-373								
EMSG$R	#5-59	8-414	8-415	8-416	9-443	10-474	10-475	11-514		
FNBDF$	#5-60	5-72								
GCL$	#5-59	10-459								
ISTAT$	#5-59	17-701								
MTRAN$	#17-701									
NTLDF$	#5-59	5-69								
NTLER$	#5-59	7-107	7-108	7-109	7-110	7-112	7-113	7-114	7-117	7-119
	7-121	7-122	7-127	7-128	7-129	7-132	7-133	7-134	7-135	7-141
	7-142	7-143	7-144	7-145	7-146	7-147	7-148	7-149	7-150	7-151
RETURN	8-293	8-350	8-371	8-409	9-438	10-464	11-509	12-571	14-629	15-656
	16-690									
SERDF$	#5-60	5-73								
SLTDF$	#5-60	5-71								
SOB	8-315	8-323								
STATE$	#5-59	17-704	#17-709	#17-713	#17-717	#17-721	#17-725	#17-729	#17-733	#17-737
	#17-740	#18-745	#18-750	#18-753	#18-757	#18-763	#18-766	19-774	#19-777	#19-782
	#19-785	#19-789	#19-793	#19-797	#19-800	#19-804	#19-808	#19-812	#19-815	#19-818
	#19-822	#19-826	#19-830	#20-839	#20-843	#20-847	#20-851	#20-855	#20-859	#20-863
	#20-868	#20-873	#20-876	#20-881	#20-885	#20-888	#20-891	#20-894	21-904	#21-906
	#21-910	#21-917	22-925	#22-927	#22-933	#22-936	#22-939	#22-942	#22-945	#22-948
	#22-951	#22-954	#22-957	#22-960	23-968	#23-987	#23-989	#23-991	#23-993	#23-995
	#23-997	#23-999	#23-1001	#23-1003	#23-1005	#23-1007	#23-1009	#23-1011	#23-1013	24-1022
	#24-1024	#24-1026	#24-1032	#24-1034	#24-1037	#24-1039	#24-1042	25-1050	#25-1052	#25-1054
	#25-1056	#25-1058	#25-1060	#25-1062	#25-1065	#25-1070	#25-1073	#25-1075	#25-1077	#25-1079
	#25-1082	#25-1085	#25-1088	#25-1091	#25-1094	#25-1097	#25-1100	26-1110	#26-1112	#26-1114
	#26-1116	#26-1119	#26-1130	27-1137	#27-1140	#27-1143	#27-1147	#27-1150	#27-1153	#27-1156
	#27-1159	#27-1162	#27-1165	#27-1169	#28-1178	#28-1181	#28-1184	#28-1187	#28-1190	#28-1193
	#28-1196	#28-1199	#28-1202	#28-1205	#28-1208	#28-1211	#28-1214	#28-1217	#28-1220	#28-1223
	#28-1226	#28-1229	#28-1232	#28-1235	29-1243	#29-1246	#29-1249	#29-1252	#29-1255	#29-1258
	#29-1261	30-1269	#30-1272	#30-1275	#30-1278	#30-1281	#30-1284	#30-1292	#30-1295	31-1303
	#32-1307									
SWSTK$	8-345	8-367								
TRAN$	#5-59	#17-705	#17-706	#17-707	#17-710	#17-711	#17-714	#17-715	#17-718	#17-719
	#17-722	#17-723	#17-726	#17-727	#17-730	#17-731	#17-734	#17-735	#17-738	#17-741

```
                                        .SBTTL  READ - READ NEXT RECORD

                                ;+
                                ; READ - READ NEXT RECORD FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       CFGSZ=RECORD LENGTH
                                ;-
000272                          READ:   GCL$    ,#CFGBF,#132.   ; READ NEXT RECORD
000312  016700  000000G                 MOV     $CLIOS,R0       ; GET GCL IOSB ADDRESS
000316  016067  000002  000336'         MOV     2(R0),CFGSZ     ; SET RECORD LENGTH
000324  105710                          TSTB    (R0)            ; SUCCESS? (CLEARS C-BIT)
000326  100401                          BMI     101$            ; IF MI, NO
000330                          10$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000332  122710  000000G         101$:   CMPB    #IE.EOF,(R0)    ; IS IT END OF FILE?
000336  000261                          SEC                     ; ASSUME THAT IT IS
000340  001773                          BEQ     10$             ; IF EQ, YES - RETURN WITH C-BIT SET
000342  122710  000000G                 CMPB    #IE.RBG,(R0)    ; IS IT BUFFER TOO SMALL?
000346  001403                          BEQ     111$            ; IF EQ, YES
000350                                  EMSG$R  1Q              ; READ FAILURE
000356                          111$:   EMSG$R  1R              ; READ OVERFLOW
```

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE	REFERENCES
LO.PAR		000010	#5-63
LO.PRI		000010	#5-63
LO.SCR		000014	#5-63
LO.STA		000015	#5-63
LO.VCT		000012	#5-63
LR.AST		000026	#5-63
LR.CIR		000002	#5-63
LR.CTL		000004	#5-63
LR.EFN		000004	#5-63
LR.LIN		000002	#5-63
LR.PRM		000030	#5-63
LR.PRO		000002	#5-63
LR.STS		000000	#5-63
LR.TCB		000002	#5-63
LR.UNT		000005	#5-63
LS.AST	=	100000	#5-63
LS.CEX	=	000001	#5-63
LS.CIR	=	000010	#5-63
LS.CXO	=	010000	#5-63
LS.CX5	=	000400	#5-63
LS.DLM	=	004000	#5-63
LS.ECH	=	001000	#5-63
LS.FDX	=	004000	#5-63
LS.HDX	=	010000	#5-63
LS.LIN	=	000003	#5-63
LS.LMC	=	000007	#5-63
LS.NTI	=	004000	#5-63
LS.ON	=	000011	#5-63
LS.OPT	=	000400	#5-63
LS.PRO	=	000002	#5-63
LS.PWF	=	040000	#5-63
LS.TOP	=	002000	#5-63
LS.UNF	=	020000	#5-63
LS.X25	=	000012	#5-63
LS.11D	=	004000	#5-63
LX.CEX	=	000004	#5-63
LX.LIN	=	000006	#5-63
LX.PRO	=	000005	#5-63
L.COST		000015	#5-64
L.CTL		000012	#5-64
L.CVA		177776	#5-64
L.DDM		000002	#5-64
L.DDS		000004	#5-64
L.DLC		000003	#5-64
L.DLM		000006	#5-64
L.DLS		000010	#5-64
L.FLG		000000	#5-64
L.KRBA		000016	#5-64
L.LEN	=	000022	#5-64
L.MPF		000022	#5-64
L.NMST		000020	#5-64
L.NSTA		000014	#5-64

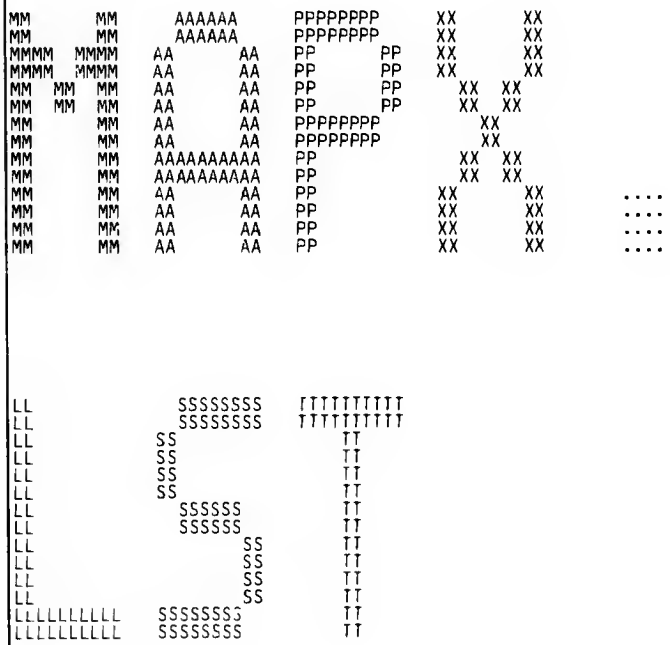
NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 4

Table of contents

5-	54	MACRO DEFINITIONS
6-	62	LOCAL DATA
8-	219	$NLDEV - SET UP DEVICE DATA STRUCTURES
10-	348	DVSYM - DEFINE SPECIAL DATA STRUCTURE SYMBOLS
11-	419	DVLBL - OPEN FILE AND READ LABEL BLOCK
12-	466	DVALL - ALLOCATE DEVICE TABLE SPACE
13-	515	DVXFR - TRANSFER DEVICE TABLES INTO CORE
14-	624	RVAD - RELOCATE AND VALIDATE DATA STRUCTURES
14-	625	VAD - VALIDATE DATA STRUCTURES
15-	643	$NLCDD - CALL DRIVER AT POWERFAIL ENTRY
16-	685	DVERR - PRINT ERROR MESSAGE
17-	723	DVDEA - DEALLOCATE DATA STRUCTURES AND CLOSE FILE
17-	724	FNDEA - PRINT FILE NAME AND DEALLOCATE
17-	725	$FNDEQ - PRINT FILE NAME AND CLOSE FILE
18-	750	REP4A - REPLACE XXTAB NAME AND SYMBOL NAME
18-	751	REP4A1 - REPLACE XXX NAME AND SYMBOL NAME
19-	797	REP9 - REPLACE DEVICE NAME
20-	814	REPDV

```
                                        .SBTTL  DVALL - ALLOCATE DEVICE TABLE SPACE

                                ;+
                                ; DVALL - ALLOCATE DEVICE TABLE SPACE FROM THE POOL
                                ;
                                ; THIS ROUTINE IS CALLED TO ALLOCATE SPACE FOR THE DEVICE TABLES FROM THE
                                ; CORE POOL. THE AMOUNT TO ALLOCATE IS DEFINED BY THE VALUE OF $XXEND FROM
                                ; THE STB FILE. THE SYMBOL VALUES FROM THE STB FILE ARE ADJUSTED BY ADDING
                                ; THE ADDRESS OF THE ALLOCATED DATA.
                                ;
                                ; INPUTS:
                                ;       LENGTH=BYTE COUNT NEEDED
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=BUFFER ADDRESS (IF SUCCESSFUL)
                                ;       $LLEN=DEVICE TABLES LENGTH
                                ;-
000636  005000                  DVALL:  CLR     R0              ; ZERO THE BUFFER ADDRESS
000640  016701  000136'                 MOV     LENGTH,R1       ; GET BYTE COUNT NEEDED
000644                                  SWSTK$  20$             ;* ENTER KERNEL MODE
000650                                  CALL    @ALOCB          ;* TRY TO ALLOCATE FROM THE POOL
000654  103402                          BCS     10$             ;* IF CS, FAILURE
000656  010066  000002                  MOV     R0,2(SP)        ;* SET USER MODE R0
000662                          10$:    RETURN                  ;* BACK TO USER MODE
000664  005700                  20$:    TST     R0              ; DID WE SUCCEED?
000666  001427                          BEQ     101$            ; IF EQ, NO
000670  010102                          MOV     R1,R2           ; COPY ALLOCATION SIZE
000672  062702  000003                  ADD     #3,R2           ; ROUND IT UP TO NEXT MULTIPLE OF 4
000676  042702  000003                  BIC     #3,R2           ; IN MEMORY OF THE RPO* SERIES
000702  010267  000000G                 MOV     R2,$LLEN        ; SET IMAGE FILE TRANSFER SIZE
000706  060001                          ADD     R0,R1           ; CALCULATE END OF TABLES-1
000710  005301                          DEC     R1              ;    ...
000712  010167  000174'                 MOV     R1,TABEND       ; SAVE IT FOR LATER
000716  166700  000170'                 SUB     BASE,R0         ; ADJUST FOR TKB BASE ADDRESS
000722  012701  000140'                 MOV     #DBLK2,R1       ; START AT SECOND CONTROL BLOCK
000726  060061  000010          30$:    ADD     R0,10(R1)       ; ADJUST DEVICE TABLE SYMBOL
000732  011101                          MOV     (R1),R1         ; GET NEXT CONTROL BLOCK ADDRESS
000734  001374                          BNE     30$             ; IF NE, NOT AT END
000736  066700  000170'                 ADD     BASE,R0         ; RESTORE ALLOCATION ADDRESS
000742  000241                          CLC                     ; INDICATE SUCCESS
000744                                  RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000746                          101$:   EMSG$R  2L              ; ALLOCATION FAILURE
```

```
**FILE**ID**MAPX
```




```
MAPX    MACRO V05.03b Sunday 30-Jun-85 00:00
Table of contents
```

Symbol table

```
A$$CHK= 000000          C.TYPR= 000000          F.HIBK= 000004          M$$NET= 000000          S.FDB = 000140
A$$CPS= 000000          DEFFLG= 000233          F.LUN = 000042          M$$OVR= 000000          S.FNAM= 000006
A$$PRI= 000000          DFLTNB  000426R     002 F.MBCT= 000054          N$$ACC= 000001          S.FNB = 000036
A$$TRP= 000000          D$$BUG= 177514          F.MBCT= 000055          N$$BUF= 000001          S.FNBW= 000017
A.LULU= 000002          D$$ISK= 000000          F.MBFG= 000056          N$$LDV= 000001          S.FNTY= 000004
A.LUNA= 000004          D$$L11= 000001          F.NRBD= 000024          N$$MCP= 000001          S.FTYP= 000002
A.LUNU= 000006          D$$YNC= 000000          F.NREC= 000030          N$$MLL= 000001          S.NFEN= 000020
BUFFER  000472R     002 D$$YNM= 000000          F.OVBS= 000030          N$$MOV= 000010          T$$KMG= 000000
BUFLEN= 000001          EF    = 000001          F.RACC= 000016          N$$NCT= 000001          T$$MIN= 000000
CL.AST= 004000          E$$XPR= 000000          F.RATT= 000001          N$$PEM= 000001          V$$CTR= 001000
CL.ATT= 000200          FX.DIR= 000004          F.RCNM= 000034          N.ANM1= 000002          W.TSEF= 000002
CL.CAS= 020000          FX.DLN= 000003          F.RCTL= 000017          N.ANM2= 000024          X$$DBT= 000000
CL.CMT= 000001          FX.DMX= 000002          F.RSIZ= 000002          N.DID = 000024          $ASTLV  000210RG    002
CL.EIF= 000040          FX.LEN= 000000          F.RTYP= 000000          N.DVNM= 000032          $CLASN  000162RG    002
CL.EXN= 040000          F$$LVL= 000001          F.SEQN= 000100          N.FID = 000000          $CLAST  000172RG    002
CL.EXT= 000010          F.ACTL= 000076          F.SPDV= 000072          N.FNAM= 000006          $CLBUF= 000310RG    002
CL.IND= 000002          F.ALOC= 000040          F.SPUN= 000074          N.FTYP= 000014          $CLCSI  000212RG    002
CL.LCC= 000020          F.BBFS= 000062          F.STBK= 000036          N.FVER= 000016          $CLEVL  000206RG    002
CL.NPR= 000400          F.BDB = 000070          F.UNIT= 000136          N.NEXT= 000022          $CLFDB  000266RG    002
CL.OPN= 000100 G        F.BGBC= 000057          F.URBD= 000020          N.STAT= 000020          $CLFLG  000464RG    002
CL.RNE= 000004          F.BKDN= 000026          F.VBN = 000064          N.UNIT= 000034          $CLIOS= 000132RG    002
CL.XIT= 100000          F.BKDS= 000020          F.VBSZ= 000060          P$$P45= 000000          $CLMCR  000000RG    002
CS.DIF= 000002          F.BKEF= 000050          G$$TPP= 000000          P$$WRD= 000000          $CLPMT  000466RG    002
CS.DVF= 000004          F.BKP1= 000051          G$$TSS= 000000          Q$$OPT= 000010          $CLQIO  000122RG    002
CS.EQU= 000040          F.BKST= 000024          G$$TTK= 000000          Q.IOAE= 000012 G        $CLRTN= ****** GX
CS.INP= 000001          F.BKVB= 000064          G$$WRD= 000000          Q.IOEF= 000006          $CLSBK  000176RG    002
CS.MOR= 000020          F.CHR = 000075          G.MCRB= 000002 G        Q.IOFN= 000002 G        $CLSEF  000156RG    002
CS.NMF= 000001          F.CNTG= 000034          I$$RAR= 000000          Q.IOLU= 000004          $CLSIZ= 000306RG    002
CS.OUT= 000002          F.DFNB= 000046          I$$RDN= 000000          Q.IOPL= 000014 G        $CLSRV  000470RG    002
CS.WLD= 000010          F.DSPT= 000044          K$$CNT= 177546          Q.IOPR= 000007          $CLTBF  000202RG    002
C$$CKP= 000000          F.DVNM= 000134          K$$CSR= 177546          Q.IOSB= 000010 G        $CLTSZ  000204RG    002
C$$DRE= 000400          F.EFBK= 000010          K$$LDC= 000000          R$$DER= 000000          $CLWEF  000152RG    002
C$$RSH= 177564          F.EFN = 000050          K$$TPS= 000074          R$$K11= 000001          $PSHPT  000174RG    002
C.CMLD= 000002          F.EOBB= 000032          LD$LP = 000000          R$$SND= 000000          $$$ARG= 000003
C.DEVD= 000006          F.ERR = 000052          L$$ASG= 000000          R$$T1M= 000000          $$$OST= 000010
C.DIRD= 000012          F.EXT = 000072          L$$DRV= 000000          S$$WRG= 000000          $$$T1 = 000002
C.DSDS= 000006          F.FACC= 000043          L$$P11= 000001          S$$YSZ= 007600          $$$T2 = 000166R     002
C.FILD= 000016          F.FFBY= 000014          L$$11R= 000000          S.ANM1= 000014          .FSRCB= ****** G
C.MKW1= 000024          F.FLG = 000074          M$$CRB= 000124          S.ANM2= 000005          ...PC1= 000266R     002
C.MKW2= 000026          F.FNAM= 000110          M$$CRX= 000000          S.BFHD= 000020          ...PC2= 000464R     002
C.SIZE= 000054          F.FNB = 000102          M$$FCS= 000000          S.ETEF= 000002          ...PC3= 000266R     002
C.STAT= 000001          F.FTYP= 000116          M$$MGE= 000000          S.FATT= 000016          ...TPC= 000020
C.SWAD= 000022          F.FVER= 000120

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
$$$GCL  000474    002   (RW,I,LCL,REL,CON)
$$FSR1  001020    003   (RW,D,GBL,REL,OVR)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13638 Words  ( 54 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
```


Symbol table

```
                                        .SBTTL  $SCEX - TRANSFER COMM EXEC INTO CORE

                                ;+
                                ; $SCEX - TRANSFER COMM EXEC INTO CORE
                                ;
                                ; INPUTS:
                                ;       .HOME=NETLDR DATA BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       .CXSYM=PROCESS VALIDATION SYMBOL DEFINITIONS
                                ;       .CXPCB=COMM EXEC PCB ADDRESS
                                ;       .CXALL=CETAB ALLOCATION ADDRESS (RSX-11M+)
                                ;       .CXLEN=CETAB ALLOCATION LENGTH (RSX-11M+)
                                ;       FE.CEX OF $FMASK SET IF SUCCESSFUL
                                ;
                                ;-
000000                          $$CEX:: CALL    CESYM           ; LOOK FOR SPECIAL SYMBOLS
000004  103540                        BCS     10$             ; IF CS, ERROR

                                        .IF NDF R$$M
000006  012700  000000G                 MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE SPEC
000012  012701  000006'                 MOV     #CEXNAM,R1      ; FOR [130,5X]CEX.STB
000016  005002                          CLR     R2              ;       ...
000020                                  CALL    $AZFS           ;       ...
000024                                  CALL    $PRIO           ; TRANSFER COMM EXEC INTO CORE
000030  103021                        BCC     2$              ; IF SUCCESS - BRANCH

000032                                  SWSTK$  1$
000036  016700  000000G                 MOV     .HOME,R0        ; GET DATA BLOCK BIAS
000042                                  CALL    $MAPX           ; MAP TO BLOCK
000046  011267  000002'                 MOV     (R2),NTPT       ; SAVE POOL.. PCB ADDRESS
000052                                  RETURN
000054  010046                  1$:     MOV     R0,-(SP)        ; SAVE ERROR CODE
000056  016700  000002'                 MOV     NTPT,R0         ; STORE PCB ADDRESS
000062  001402                          BEQ     3$              ; IF NONE - BRANCH
000064                                  CALL    $XPAR           ; RELEASE PARTITION
000070  012600                  3$:     MOV     (SP)+,R0        ; RESTORE ERROR CODE
000072  000511                          BR      ISKERR          ; IF CS, ERROR
000074  010167  000036'          2$:     MOV     R1,CEXPCB       ; SAVE COMM EXEC PCB ADDRESS
000100  016700  000000G                 MOV     CEAVL,R0        ; GET ALLOCATION POINTER ADDRESS
000104  001427                          BEQ     5$              ; IF UNDEFINED, DO NOTHING
000106  026027  177776  000003          CMP     -2(R0),#3       ; REASONABLENESS CHECK
000114  001075                          BNE     10$             ;       ...
000116  016102  000000G                 MOV     P.REL(R1),R2    ; GET PARTITION STARTING ADDRESS
000122  066102  000000G                 ADD     P.BLKS(R1),R2   ; ADD PARTITION LENGTH

000126                                  ASL$    6,R2            ; CONVERT FROM BLOCKS TO BYTES
000142  010001                          MOV     R0,R1           ; COPY POINTER ADDRESS
000144  022121                          CMP     (R1)+,(R1)+     ; SKIP THE NEXT TWO WORDS
000146  064001                          ADD     -(R0),R1        ; ROUND UP <POINTER ADDRESS+4>
000150  042001                          BIC     (R0)+,R1        ;       ...
000152  160102                          SUB     R1,R2           ; GIVING POOL LENGTH
000154  001403                          BEQ     5$              ; IF ZERO LENGTH, DO NOTHING
000156  010110                          MOV     R1,(R0)         ; SET ADDRESS OF FIRST SEGMENT
000160  005021                          CLR     (R1)+           ; NO NEXT SEGMENT
```

```
                                        .SBTTL  RDBS - ALLOCATE RDBS

                                ;+
                                ;
                                ; RDBS - CARVE RDB'S OUT OF NETWORK POOL
                                ;
                                ; INPUTS:
                                ;       R4      - BIAS OF BASE OF POOL
                                ;       R5      - VIRTUAL ADDRESS OF BASE OF POOL
                                ;
                                ; OUTPUTS:
                                ;       R4      - BIAS OF BASE OF NEXT FREE SEGMENT OF POOL
                                ;       R5      - VIRTUAL ADDRESS OF NEXT FREE BLOCK
                                ;
                                ;-

002032                          RDBS:
002032  017701  000000G                 MOV     @RDBNM,R1               ; NUMBER OF RDBS TO ALLOCATE
002036  017703  000000G                 MOV     @RDBSZ,R3               ; NUMBER OF BYTES/RDB
002042  062703  000000G                 ADD     #CRC$OV,R3              ;   ADD IN CRC OVERHEAD
002046  067705  000000G         20$:    ADD     @MAXOV,R5               ; SKIP DLC OVERHEAD IN RDB

002052                                  SWSTK$  30$                     ;* ENTER KERNEL MODE

002056  010400                          MOV     R4,R0                   ;* COPY BIAS
002060                                  CALL    $MAPX                   ;* MAP TO THE BUFFER
002064  042705  160000                  BIC     #160000,R5              ;* REMOVE APR BITS FROM ADDRESS
002070  050205                          BIS     R2,R5                   ;* THEN INCLUDE $MAPX APR
002072  016704  000000G                 MOV     RDBLH,R4                ;* GET ADDRESS OF $RDBLH
002076  012425                          MOV     (R4)+,(R5)+             ;* INITIALIZE BIAS
002100  011415                          MOV     (R4),(R5)               ;* AND VIRTUAL ADDRESS

002102                                  RETURN                          ;* RETURN FROM SYSTEM STATE

002104                          30$:
002104  010477  000000G                 MOV     R4,@RDBLH               ; SET UP RDB BIAS
002110                                  SAVRG   <R4>                    ; SAVE BIAS
002112  016704  000000G                 MOV     RDBLH,R4                ; GET $RDBLH ADDRESS
002116  062704  000002                  ADD     #2,R4                   ; POINT TO SECOND WORD OF LISTHEAD
002122  010514                          MOV     R5,(R4)                 ; STORE LISTHEAD VIRTUAL ADDRESS
002124                                  RESRG   <R4>                    ; RTORE BIAS
002126                                  CALL    GTBUF                   ; GET MAPPING FOR NEXT BUFFER
002132  005301                          DEC     R1                      ; LOOP TILL
002134  001344                          BNE     20$                     ;  DONE.
002136                                  RETURN
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 6

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES					
$ERRQY	= ******	GX	8-482					
$ERRXQ	= ******	GX	14-865					
$ERRZA	= ******	GX	7-376	11-666				
$ERR44	= ******	GX	6-256					
$ERR45	= ******	GX	6-257					
$ERR46	= ******	GX	6-258					
$ERR47	= ******	GX	6-259					
$ERR48	= ******	GX	8-481					
$ERR51	= ******	GX	6-264					
$ERR53	= ******	GX	6-266					
$ERR54	= ******	GX	6-267					
$ERR55	= ******	GX	6-268					
$ERR56	= ******	GX	6-269					
$ERR57	= ******	GX	6-270					
$ERR58	= ******	GX	6-271					
$ERR59	= ******	GX	6-272					
$ERR60	= ******	GX	6-273					
$ERR61	= ******	GX	6-274					
$ERR62	= ******	GX	6-275					
$ERR63	= ******	GX	6-276					
$ERR64	= ******	GX	6-277					
$ERR65	= ******	GX	6-278					
$ERR66	= ******	GX	6-279					
$ERR67	= ******	GX	6-280					
$ERR68	= ******	GX	6-281					
$FLAGS	= ******	GX	11-658					
$FRKHD	= ******	GX	6-181					
$HEADR	= ******	GX	6-186	18-1014				
$MAPX	= ******	GX	7-316	10-552	13-753	13-765	13-782	15-892	16-939
$PRCLN	000004	RG	#6-115					
$PREXT	000004	RG	#6-116					
$PRINC	000004	RG	#6-117					
$PRIO	= ******	GX	7-311					
$PRLEN	000004	RG	#6-118					
$PRLLN	000004	RG	#6-119					
$RSXNM	= ******	GX	*8-398	*8-448	8-449			
$SLTA	= ******	GX	18-1022					
$STKDP	= ******	GX	6-191					
$SYMBL	= ******	GX	8-411	8-418	8-425			
$XPAR	= ******	GX	7-322					
$$HTIM	= ******	GX	18-1048					
$$LTIM	= ******	GX	18-1049					
.ALLOC	= ******	GX	13-745					
.AMAXC	= ******	GX	11-618					
.AMAXH	= ******	GX	11-619					
.BASEB	= 140000		#5-68	13-794	17-983			
.CCBNM	= ******	GX	13-718	13-723				
.CXALL	= ******	GX	*10-577					
.CXCSR	= ******	GX	10-553					
.CXLEN	= ******	GX	*10-578					
.HOME	= ******	GX	7-315	10-548	11-622			
.LLCTB	000276	R	6-240	#6-243	11-605			

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 7

```
                                                BCC     40$             ; IF CC, YES
                                                CALL    READ            ; READ NEXT RECORD
                                                BCC     80$             ; IF CC, NOT EOF
                                        90$:    FMSG$R  1Y              ; DDM$DF MISSING (EOF)

                                        ;
                                        ; ALL MISSING LINE INFO HAS BEEN FOUND
                                        ;
000450  103311                          100$:   BIT     #FG.X2P,CFGFLG  ; DO WE NEED X2P$DF?
000452                                          BEQ     190$            ; BR IF NO
000456  103372                                 CLR     UCNT            ; INITIALIZE UNIT COUNT
000460                                  110$:   CALL    READ            ; READ NEXT RECORD
000502                                          BCS     171$            ; BR IF END OF FILE
000506  103523                                 CALL    $QX2P           ; IS IT X2P$DF?
000510                                          BCS     110$            ; BR IF NO - KEEP LOOKING
000514  103772                                 CMP     $$MUX,UCNT      ; HAVE ALL X2P$DF'S BEEN SEEN?
000516  026767  000000G 000142'                BNE     110$            ; IF NE, NO
000524  001366                                 BR      190$            ; ELSE FINISHED
000526  000462

                                        ;
                                        ; LOOK FOR DLC$DF OR DDM$DF (PROCESS INCREMENT NEEDED)
                                        ;
000530                                  120$:   CALL    READ            ; READ NEXT RECORD
000534  103751                                 BCS     90$             ; IF CS, EOF
000536                                          CALL    $QDDM           ; IS IT DLC$DF OR DDM$DF ?
000542  103772                                 BCS     120$            ; IF CS, NO - KEEP LOOKING
000544  000453                                 BR      190$            ; ELSE, LEAVE


                                        ;
                                        ; LOOK FOR LLC$DF
                                        ;
000546                                  130$:   CALL    READ            ; READ NEXT RECORD
000552  103473                                 BCS     151$            ; IF CS, EOF
000554                                          CALL    QLLC            ; IS IT LLC$DF?
000560  103772                                 BCS     130$            ; IF CS, NO
000562  005067  000000G                        CLR     $FEATR          ; CLEAR THE FEATURES WORD
000566                                  140$:   CALL    READ            ; READ NEXT RECORD
000572  103434                                 BCS     180$            ; IF CS, END OF FILE
000574                                          CALL    $QFEA           ; IS IT FEA$DF?
000600  103016                                 BCC     160$            ; BR IF YES
000602  032767  000002  000136'                BIT     #FG.X3P,CFGFLG  ; IS X3P$DF NEEDED?
000610  001766                                 BEQ     140$            ; BR IF NO
000612                                          CALL    $QX3P           ; IS IT X3P$DF?
000616  103763                                 BCS     140$            ; BR IF NO

                                        ;
                                        ; X3P$DF FOUND
                                        ;
000620                                  150$:   CALL    READ            ; READ NEXT RECORD
000624  103423                                 BCS     190$            ; BR IF END OF FILE
000626                                          CALL    $QFEA           ; IS IT FEA$DF
000632  103772                                 BCS     150$            ; BR IF NO
000634  000417                                 BR      190$            ; ELSE FINISHED

                                        ;
                                        ; FEA$DF FOUND
                                        ;
000636  032767  000002  000136'         160$:   BIT     #FG.X3P,CFGFLG  ; IS X3P$DF NEEDED
000644  001413                                 BEQ     190$            ; BR IF NO
000646                                  170$:   CALL    READ            ; READ NEXT RECORD
000652  103436                                 BCS     161$            ; BR IF EOF - X3P$DF NOT FOUND
```

```
                                        .SBTTL  ADD$DF ACTION ROUTINES

                                ;
                                ; LOCAL PHYSICAL STATION ADDRESS (ADD$DF)
                                ;
001534  005767  000000G         AD.NUM: TST     .PNUMH          ; MUSTB BE BYTE VALUE
001540  001007                          BNE     101$            ; ...
001542  105767  000001G                 TSTB    .PNUMB+1        ; ...
001546  001004                          BNE     101$            ; ...
001550  116767  000000G 000000G         MOVB    .PNUMB,$$LSAD   ; STORE LOCAL PHYSICAL STATION ADDRESS
001556                                  RETURN
001560                          101$:   EMSG$R  2A              ; ILLEGAL LOCAL PHYSICAL STATION ADDRESS
```

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
$QDDM	= ******	GX	9-284	9-323	9-346						
$QDTE	= ******	GX	9-262								
$QFEA	= ******	GX	9-360	9-371							
$QPVC	= ******	GX	9-264								
$QSLT	= ******	GX	9-248	9-282	9-306						
$QSTA	= ******	GX	9-253	9-313							
$QSVC	= ******	GX	9-258								
$QUNT	= ******	GX	9-298								
$QX2P	= ******	GX	9-336								
$QX3P	= ******	GX	9-260	9-364	9-381						
$RAD50	= 000016		#16-596								
$REP.N	= ******	GX	8-170	8-171							
$RONLY	= ******		16-596	16-596	16-596						
$SLTA	= ******	GX	9-242	9-276							
$STRNG	= 000004		#16-596								
$SUBXP	= 000010		#16-596								
$TIOUT	= ******	GX	13-537	14-561							
$$DEV	= ******	GX	*24-789	*24-795							
$$EXT	= ******	GX	*20-730								
$$INC	= ******	GX	*20-721								
$$LSAD	= ******	GX	*21-747								
$$MTP	= ******	GX	*22-760								
$$MUX	= ******	GX	9-300	9-338							
$$$FLG	= 177777		#16-596								
$$$KEY	= 177777		#16-596								
.PCHAR	= ******	GX	24-789	24-795							
.PNUMB	= ******	GX	20-721	20-730	20-731	21-745	21-747	22-756	22-758	22-760	23-776
			24-803	24-811	27-882	27-896					
.PNUMH	= ******	GX	21-743	23-772	23-774	24-801	27-880	27-894			
.PSTCN	= ******	GX	20-700								
.PSTPT	= ******	GX	20-699								
.TPARS	= ******	GX	12-492								

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 6

```
                                        .SBTTL  OPEN - OPEN THE CONFIG FILE
                                ;+
                                ; OPEN - OPEN THE CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000510  012700  000000G        OPEN:   MOV     #$CFUIC,R0      ; SETUP ASCIZ FILE SPEC
000514  012701  000000'                MOV     #CFGNAM,R1      ; FOR [130,10]CETAB.MAC
000520  012702  000006'                MOV     #CFGEXT,R2      ;       ...
000524                                 CALL    $AZFS           ;       ...
000530                                 CALL    $CLQUE          ; OPEN THE CONFIG FILE
000534  105777  000000G                TSTB    @$CLIOS         ; SUCCESS? (CLEARS C-BIT)
000540  100401                         BMI     101$            ; IF MI, NO
000542                                 RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000544                          101$:   EMSG$R  1P              ; OPEN FAILURE
```

```
    891 001252                              STATE$  ZSS
    892 001252                               TRAN$  $LAMDA,,SERSLN
    893
    894 001252                              STATE$
    895 001252                               TRAN$  $LAMDA,$EXIT
    896
```

```
                                        .SBTTL  TPARS STATE TABLES -
                                ;
                                ; LOGGING STATE (LOG$ST)
                                ;
                                        STATE$  LOGST
                                         TRAN$   $NUMBR,$EXIT,L.LGST
```

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 9

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME REFERENCES

#18-746 #18-747 #18-748 #18-751 #18-754 #18-755 #18-758 #18-759 #18-760 #18-761
#18-764 #18-767 #19-775 #19-778 #19-779 #19-780 #19-783 #19-786 #19-787 #19-790
#19-791 #19-794 #19-795 #19-798 #19-801 #19-802 #19-805 #19-806 #19-809 #19-810
#19-813 #19-816 #19-819 #19-820 #19-823 #19-824 #19-827 #19-828 #19-831 #19-832
#20-840 #20-841 #20-844 #20-845 #20-848 #20-849 #20-852 #20-853 #20-856 #20-857
#20-860 #20-861 #20-864 #20-865 #20-869 #20-870 #20-874 #20-877 #20-882 #20-883
#20-886 #20-889 #20-892 #20-895 #21-905 #21-907 #21-908 #21-909 #21-911 #21-912
#21-918 #21-919 #22-926 22-931 #22-932 #22-934 #22-935 #22-937 #22-938 #22-940
#22-941 #22-943 #22-947 #22-946 #22-947 #22-949 #22-950 #22-952 #22-953 #22-955
#22-956 #22-958 #22-959 #22-961 #23-970 #23-971 #23-972 #23-973 #23-974 #23-975
#23-976 #23-977 #23-978 #23-979 #23-980 #23-981 #23-982 #23-988 #23-990 #23-992
#23-994 #23-996 #23-998 #23-1000 #23-1002 #23-1004 #23-1006 #23-1008 #23-1010 #23-1012
#23-1014 #23-1015 #24-1023 #24-1025 #24-1027 #24-1028 #24-1029 #24-1030 #24-1031 #24-1033
#24-1035 #24-1036 #24-1038 #24-1040 #24-1041 #24-1043 #25-1051 #25-1053 #25-1055 #25-1057
#25-1059 #25-1061 #25-1063 #25-1064 #25-1066 #25-1067 #25-1068 #25-1069 #25-1071 #25-1072
#25-1074 #25-1076 #25-1078 #25-1080 #25-1081 #25-1083 #25-1084 #25-1086 #25-1087 #25-1089
#25-1090 #25-1092 #25-1093 #25-1095 #25-1096 #25-1098 #25-1099 #25-1101 #26-1111 #26-1113
#26-1115 #26-1117 #26-1118 #26-1120 #26-1121 #26-1122 #26-1123 #26-1124 #26-1125 #26-1131
#27-1138 #27-1141 #27-1144 #27-1145 #27-1148 #27-1151 #27-1154 #27-1157 #27-1160 #27-1163
#27-1166 #27-1167 #27-1170 #28-1179 #28-1182 #28-1185 #28-1188 #28-1191 #28-1194 #28-1197
#28-1200 #28-1203 #28-1206 #28-1209 #28-1212 #28-1215 #28-1218 #28-1221 #28-1224 #28-1227
#28-1230 #28-1233 #29-1244 #29-1247 #29-1250 #29-1253 #29-1256 #29-1259 #29-1262 #30-1270
#30-1273 #30-1276 #30-1279 #30-1282 #30-1285 #30-1286 #30-1287 #30-1293 #30-1296 #30-1297
#31-1304

FILEID**NTLCFP

```
                                        .SBTTL  CERR,CFERR - PRINT ERROR MESSAGE

                                ;+
                                ; CERR - PRINT ERROR MESSAGE
                                ; CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
                                ; CWARN - PRINT WARNING MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET IF ERROR MESSAGE
                                ;       RETURN TO $NTICF'S CALLER OR BACK TO CALLER
                                ;-
                                        .ENABL  LSB
000364                          CWARN:
000364                                  CALL    $SAVAL          ; SAVE ALL REGISTERS
000370  005046                         CLR     -(SP)           ; WARNING MESSAGE
000372  000405                         BR      5$              ; JOIN COMMON CODE
000374                          CERR::
000374  010046                         MOV     R0,-(SP)        ; SAVE R0
000376                                  CALL    $CLDEQ          ; CLOSE THE CONFIG FILE
000402  012600                         MOV     (SP)+,R0        ; RESTORE R0
000404                          CFERR::
000404  010746                         MOV     PC,-(SP)        ; ERROR MESSAGE
000406  010005                 5$:     MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
000410  012704  000022'                 MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
000414  012503                         MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
000416                                  CALL    @(R5)+          ; DO THE FORMATTING
000420  105713                         TSTB    (R3)            ; IS IT END OF FORMAT STRING?
000422  001403                         BEQ     20$             ; IF EQ, YES
000424  112324                 10$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
000426  001376                         BNE     10$             ;       ...
000430  005304                         DEC     R4              ; SKIP THE ZERO BYTE
000432  012503                 20$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
000434  112524                 30$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
000436  001376                         BNE     30$             ;       ...
000440  005304                         DEC     R4              ; SKIP THE ZERO BYTE
000442  012700  000012'                 MOV     #ERRMSG,R0      ; GET ERROR MESSAGE ADDRESS
000446  010401                         MOV     R4,R1           ; COPY END ADDRESS
000450  160001                         SUB     R0,R1           ; GIVING LENGTH
000452                                  CALL    $TTOUT          ; PRINT IT
000456  005703                         TST     R3              ; SHOULD SECOND LINE BE PRINTED?
000460  001401                         BEQ     40$             ; IF EQ, NO
000462                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
000464  005726                 40$:    TST     (SP)+           ; ERROR OR WARNING MESSAGE ?
000466  001403                         BEQ     RTSPC           ; IF EQ, WARNING
000470  016706  000340'                 MOV     SPSAV,SP        ; RESTORE SP FROM ENTRY
000474  000261                         SEC                     ; RETURN C-BIT SET
000476                          RTSPC:: RETURN
                                        .DSABL  LSB
```

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES								
L.OWNR		000021		#5-64								
L.UNT		000013		#5-64								
MRKFL		000344	RG	#7-114								
NEXT		000352	RG	#7-119	18-613	*18-614						
OPEN		000230	R	9-250	#10-325							
PCKBCD		000700	RG	#15-513								
PDVNM	=	******	GX	17-577								
PDVTA	=	******	GX	17-576	17-588							
PORTNO		000354	RG	#7-120								
PVSTA		000402	RG	#7-145								
PV.OFF	=	000001	G	#7-146								
PV.ON	=	000002	G	#7-147								
RBRA	=	000076		#6-72								
READ		000272	R	9-251	9-255	9-262	9-278	9-290	9-294	#11-351		
RTSPC		000476	RG	8-200	8-201	8-202	8-203	8-204	8-205	8-206	8-207	8-208
				8-209	8-210	8-211	8-214	8-215	12-415	#12-418		
R$$EIS	=	******		15-524								
R$$11D	=	******		15-524								
SF.ACT	=	000200		#5-64								
SF.ENA	=	000100		#5-64								
SF.LPB	=	000004		#5-64								
SF.MFL	=	000040		#5-64								
SF.PAC	=	000020		#5-64								
SF.REA	=	000010		#5-64								
SF.SER	=	000001		#5-64								
SF.SVC	=	000002		#5-64								
SF.UNL	=	000040		#5-64								
SPACE	=	000040		#6-73								
SPSAV		000340	R	#7-112	*9-249	12-416						
STRNXT		001064	RG	#18-607								
SUBAMN	=	000000	G	#6-78								
SUBAMX	=	023417	G	#6-79								
SYNERR		000342	RG	#7-113								
S$$BAS	=	******		8-200	8-200	8-201	8-201	8-202	8-202	8-203	8-203	8-204
				8-204	8-205	8-205	8-206	8-206	8-207	8-207	8-208	8-208
				8-209	8-209	8-210	8-210	8-211	8-211	8-212	8-212	8-213
				8-213	8-214	8-214	8-215	8-215				
S.COST		000001		#5-64								
S.FLG		000000		#5-64								
S.LEN		000004		#5-64								
S.NMST		000002		#5-64								
S.OWNR		000003		#5-64								
WNDSMX	=	000177		#6-80	14-490							
WNDSZD		000364	RG	#7-129								
WNDSZM		000366	RG	#7-130								
Z.NAM	=	******	GX	17-581								
$AZFS	=	******	GX	10-328								
$CAT5	=	******	GX	14-447								
$CDTB	=	******	GX	18-612								
$CEACX	=	******	GX	16-552								
$CFUIC	=	******	GX	10-325								
$CLBUF	=	******	GX	*11-351								

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 5

```
                        .TITLE  NTLDEV - BUILD DEVICE DATA STRUCTURES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - NS: AND NX: DEVICE DATA STRUCTURE BUILDING ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                        .SBTTL  DVXFR - TRANSFER DEVICE TABLES INTO CORE

                                ;+
                                ; DVXFR - TRANSFER DEVICE TABLES INTO CORE
                                ;
                                ; THIS ROUTINE IS CALLED TO TRANSFER THE DEVICE TABLES INTO CORE, SETUP THE
                                ; INTRA-BLOCK POINTERS, AND LINK THE DCB INTO THE DEVICE LIST ($DEVHD). THE
                                ; LOCATIONS CHANGED IN THE DEVICE TABLE IMAGE ARE:
                                ;       D.UCB           DCB - POINTER TO UCB
                                ;       D.NAM           DCB - DEVICE NAME
                                ;       U.DCB           UCB - POINTER TO DCB
                                ;       U.RED           UCB - REDIRECT POINTER
                                ;       U.SCB           UCB - POINTER TO SCB
                                ;       S.LHD           SCB - I/O PACKET LISTHEAD (2 WORDS)
                                ;
                                ; INPUTS:
                                ;       R0=CORE POOL BUFFER ADDRESS
                                ;       $LLEN=DEVICE TABLES LENGTH
                                ;       DCB=DCB ADDRESS (REAL CORE ADDRESS)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000754                          DVXFR:  CALL    $READ           ; TRANSFER DEVICE TABLES INTO CORE
000760  103504                         BCS     101$            ; IF CS, FAILURE
000762  016700  000150'                MOV     DCB,R0          ; GET DCB ADDRESS
000766  016760  000000G 000000G        MOV     $$DEV,D.NAM(R0) ; SETUP DEVICE NAME
000774  005001                         CLR     R1              ; GET FIRST UNIT NUMBER
000776  156001  000000G                BISB    D.UNIT(R0),R1   ;   ...
001002  005003                         CLR     R3              ; AND LAST UNIT NUMBER
001004  156003  000001G                BISB    D.UNIT+1(R0),R3 ;   ...
001010  160103                         SUB     R1,R3           ; CALCULATE NUMBER OF UNITS
001012  005203                         INC     R3              ;   ...
001014  003474                         BLE     111$            ; IF NOT POSITIVE, ILLEGAL
001016  010367  000172'                MOV     R3,UNITS        ; SAVE IT FOR LATER
001022  010002                         MOV     R0,R2           ; COPY DCB ADDRESS
001024  062702  000000G                ADD     #D.UCB,R2       ; GET D.UCB ADDRESS
001030                                 CALL    VAD             ; VALIDATE IT
001034  103467                         BCS     121$            ; IF CS, ILLEGAL
001036  011202                         MOV     (R2),R2         ; GET FIRST UCB ADDRESS
001040                                 CALL    RVAD            ; RELOCATE AND VALIDATE IT
001044  103466                         BCS     131$            ; IF CS, ILLEGAL
001046  010260  000000G                MOV     R2,D.UCB(R0)    ; SET FIRST UCB ADDRESS
001052  010201                 10$:    MOV     R2,R1           ; PUT UCB ADDRESS IN R1
001054  062702  000000G                ADD     #U.SCB,R2       ; GET U.SCB ADDRESS
001060                                 CALL    VAD             ; VALIDATE IT
001064  103466                         BCS     141$            ; IF CS, ILLEGAL
001066  011202                         MOV     (R2),R2         ; GET SCB ADDRESS
001070                                 CALL    RVAD            ; RELOCATE AND VALIDATE IT
001074  103465                         BCS     151$            ; IF CS, ILLEGAL
001076  010061  000000G                MOV     R0,U.DCB(R1)    ; SETUP DCB ADDRESS
001102  010161  000000G                MOV     R1,U.RED(R1)    ; REDIRECT POINTER
001106  010261  000000G                MOV     R2,U.SCB(R1)    ; SCB ADDRESS
                                       .IF DF  R$$MPL
                                       ADD     SKS5,R2         ; POINT TO S.KS5 OFFSET
                                       MOV     $BIAS,(R2)      ; STORE THE BIAS FOR THIS PROCESS IN THE SCB
```

F 1

MAPX MACRO V05.03b Sunday 30-Jun-85 00:00
Table of contents

5- 61 $MAPX - MAP TO EXTENDED MEMORY
6- 97 $MAPXP - MAP TO EXTENDED POOL

G 1

F 2

NTL - IDENT MODULE MACRO V05.03b Saturday 29-Jun-85 00:22 Page 6-2
Symbol table

Operating system: RSX-11M/PLUS

Elapsed time: 00:00:18.27
SY:NTL.V2,[132,134]NTL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTL

G 2

NTL CREATED BY MACRO ON 29-JUN-85 AT 00:22 PAGE 1
SYMBOL CROSS REFERENCE CREF 04.00

```
000162  010211                          MOV     R2,(R1)         ; SET SEGMENT LENGTH
                                        .ENDC
000164  012700  000000G         5$:     MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE SPEC
000170  012701  000012'                 MOV     #CETNAM,R1      ; FOR [XXX,5X]CETAB.STB
000174  005002                          CLR     R2              ;       ...
000176                                  CALL    $AZFS           ;       ...
000202                                  CALL    $CTBIO          ; TRANSFER CETAB INTO CORE
000206  103443                          BCS     TSKERR          ; IF CS, ERROR

000210                                  CALL    NLBLK           ; SETUP NETLDR DATA BLOCK
000214                                  CALL    DECBLK          ; SETUP DECNET HOME BLOCK POINTER
000220  103432                          BCS     10$             ; IF FAILURE - BRANCH
000222                                  CALL    CEPTR           ; RELOCATE CETAB ADDRESSES
000226  005000                          CLR     R0              ;
000230                                  CALL    ELTIME          ; CALCULATE TIME SINCE MIDNIGHT
000234  016701  000000G                 MOV     ZTIME,R1        ; GET $ZTIME ADDRESS
000240  010021                          MOV     R0,(R1)+        ; INITIALIZE FOR SYSTEM COUNTERS
000242  010011                          MOV     R0,(R1)         ; INITIALIZE $ZTIME+2 FOR NODE COUNTERS

000244  103420                          BCS     10$             ; IF CS, FAILURE
000246  052777  000000G 000000G        BIS     #FE.CEX,@FMASK  ; INDICATE COMM EXEC IS LOADED
000254  016777  000000G 000000G        MOV     CEPWR,@NETPF    ; SET UP ADDRESS OF POWERFAIL RECOVERY ROUTINE
000262  012701  000006                  MOV     #6,R1           ; GET LENGTH OF NODE NAME
000266  017700  000000G                 MOV     @DECPT,R0       ; POINT TO NODE NAME
000272  062700  000006                  ADD     #D$LNAM,R0      ;       ...
000276  112720  000040          7$:     MOVB    #' ,(R0)+       ; INIT TO SPACES IN CASE PSI ONLY SYSTEM
000302                                  SOB     R1,7$           ; ...
000306                          10$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000310                          101$:   EMSG$R  ZA              ; INCONSISTENT COMM EXEC IMAGE
000316  016000  000342'         TSKERR: MOV     ETAB2-2(R0),R0  ; GET ERROR MESSAGE CONTROL BLOCK ADDRESS
000322                                  CALLR   @(R0)+          ; PRINT ERROR MESSAGE
```

```
                                        .SBTTL  SDBS - ALLOCATE SDBS

                                ;+
                                ;
                                ; SDBS - CARVE SDB'S OUT OF NETWORK POOL
                                ;
                                ; INPUTS:
                                ;       R4      - BIAS OF BASE OF FREE SPACE IN POOL
                                ;       R5      - VIRTUAL ADDRESS OF BASE OF FREE SPACE
                                ;
                                ; OUTPUTS:
                                ;       R4      - BIAS OF BASE OF FREE SPACE
                                ;       R5      - VIRTUAL ADDRESS OF FREE SPACE
                                ;
                                ;-

002140                          SDBS:
002140  017701  000000G                 MOV     @SDBNM,R1               ; NUMBER OF SDBS TO ALLOCATE
002144  001440                          BEQ     50$                     ; IF ZERO, NO WORK TO DO
002146  017703  000000G                 MOV     @SDBSZ,R3               ; NUMBER OF BYTES/SDB
002152  062703  000000G                 ADD     #CRC$OV,R3              ; ADD CRC BYTES
002156  067705  000000G         10$:    ADD     @MAXOV,R5               ; SKIP DLC OVERHEAD

002162                                  SWSTK$  40$                     ;* ENTER KERNEL MODE

002166  010400                          MOV     R4,R0                   ;* COPY BIAS
002170                                  CALL    $MAPX                   ;* MAP TO THE BUFFER
002174  042705  160000                  BIC     #160000,R5              ;* REMOVE APR BITS FROM ADDRESS
002200  050205                          BIS     R2,R5                   ;* THEN INCLUDE $MAPX APR
002202  016704  000000G                 MOV     SDBLH,R4                ;* GET $SDBLH ADDRESS
002206  012425                          MOV     (R4)+,(R5)+             ;* INITIALIZE BIAS
002210  011415                          MOV     (R4),(R5)               ;* AND VIRTUAL ADDRESS

002212                                  RETURN                          ;* RETURN FROM SYSTEM STATE

002214  010477  000000G         40$:    MOV     R4,@SDBLH               ; SET UP SDB BIAS
002220                                  SAVRG   <R4>                    ; SAVE BIAS
002222  016704  000000G                 MOV     SDBLH,R4                ; GET $SDBLH ADDRESS
002226  062704  000002                  ADD     #2,R4                   ; POINT TO SECOND WORD OF LISTHEAD
002232  010514                          MOV     R5,(R4)                 ; SET UP LISTHEAD VIRTUAL ADDRESS
002234                                  RESRG   <R4>                    ; RESTORE BIAS

002236                                  CALL    GTBUF                   ; GET DOUBLE-WORD ADDRESS OF NEXT BUFFER
002242  005301                          DEC     R1                      ; LOOP TILL
002244  001344                          BNE     10$                     ;  DONE.
002246                          50$:    RETURN
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 7

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES		
.MAXC	= ****** GX	11-613		
.MAXH	= ****** GX	11-614	11-615	
.MXPRO	= ****** GX	13-730		
.NA	= ****** GX	11-616		
.NBEA	= ****** GX	11-617		
.NBRA	= ****** GX	18-1050		
.NN	= ****** GX	11-611		
.NOMAP	= ****** GX	13-761		
.NTPSZ	= ****** GX	13-789		
.PCB	= ****** GX	13-733		
.PDVTB	000252 R	#6-237	8-424	11-603
.PUMR	= ****** GX	13-778		
.RDBNM	= ****** GX	13-726	13-727	13-728
.ROUTB	= ****** GX	11-620		
.ROUTM	= ****** GX	11-612		
.RPRI	= ****** GX	18-1051		
.SDBNM	= ****** GX	13-729		
.SLTMB	000264 R	6-237	#6-240	11-604
.UMRFL	= ****** GX	13-759		

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 8

```
                                        CALL    $QX3P           ; IS IT X3P$DF?
000654
000660  103772                          BCS     170$            ; BR IF NO
000662  000404                          BR      190$            ; ELSE FINISHED
                                ;
                                ; END OF FILE FOUND (NO X3P$DF OR FEA$DF)
                                ;
000664  032767  000002  000136' 180$:   BIT     #FG.X3P,CFGFLG  ; X3P$DF NEEDED?
000672  001026                          BNE     161$            ; BR IF YES - ERROR
                                ;
                                ; ALL NEEDED PARAMETERS SEEN
                                ;
000674                          190$:   CALL    $CLDEQ          ; CLOSE CONFIG FILE
000700  000241                          CLC                     ; INDICATE SUCCESS
000702                                  RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000704                          101$:   EMSG$R  1S              ; UNT$DF MISSING
000712                          111$:   EMSG$R  1U              ; SLT$DF MISSING (EOF)
000720                          121$:   EMSG$R  1V              ; CNT$DF MISSING (EOF)
000726                          131$:   EMSG$R  1W              ; UNT$DF MISSING (EOF)
000734                          141$:   EMSG$R  1X              ; STA$DF MISSING (EOF)
000742                          151$:   EMSG$R  1Z              ; LLC$DF MISSING (EOF)
000750                          161$:   EMSG$R  3A              ; X3P$DF MISSING (EOF)
000756                          171$:   EMSG$R  3B              ; X2P$DF MISSING (EOF)
000764                          181$:   EMSG$R  3C              ; SVC$DF MISSING (EOF)
```

```
                                        .SBTTL PLT$DF ACTION ROUTINES

                                ;
                                ; NUMBER OF CHANNELS TO ALLOCATE FOR AN NI INTERFACE
                                ;
001566  005767  000000G        PL.CHN: TST     .PNUMB          ; PARAMETER MUST BE NON-ZERO
001572  001410                        BEQ     101$            ; IF ILLEGAL - BRANCH
001574  022767  000077  000000G       CMP     #63.,.PNUMB     ; CHECK HIGH LIMIT
001602  103404                        BLO     101$            ; IF ILLEGAL - BRANCH
001604  016767  000000G 000000G       MOV     .PNUMB,$$MTP    ; COPY THE NUMBER OF STATIONS

001612                                RETURN

001614                          101$:   EMSG$R  1A              ; ILLEGAL PLT$DF PARAMETER
```

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 6

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-62									
CALL	9-234	9-237	9-245	9-248	9-250	9-253	9-255	9-258	9-260	9-262
	9-264	9-279	9-282	9-284	9-289	9-292	9-296	9-298	9-304	9-306
	9-311	9-313	9-315	9-317	9-323	9-325	9-334	9-336	9-344	9-346
	9-353	9-355	9-358	9-360	9-364	9-369	9-371	9-379	9-381	9-392
	10-423	10-424	11-446	12-492	13-519	13-524	13-537	13-540	14-558	26-869
	26-874									
CALLR	14-561									
DBGTP$	#16-596	#16-605	#16-606	#16-607	#16-610	#16-611	#16-614	#16-615	#16-621	#18-654
	#18-657	#18-662	#18-665	#18-668	#18-671	#18-674				
DHBDF$	#5-62	5-65								
EMSG$R	#5-62	9-327	9-399	9-400	9-401	9-402	9-403	9-404	9-405	9-406
	9-407	10-432	11-461	11-462	12-501	20-735	21-749	22-764	24-805	
GCL$	#5-62	11-446								
ISTAT$	#5-62	16-596								
MTRAN$	#16-596									
NTLER$	#5-62	8-170	8-171	8-172	8-173	8-174	8-175	8-176	8-177	8-178
	8-179	8-180	8-181	8-182	8-183	8-184	8-185	8-186	8-187	8-188
	8-189	8-190	8-191	8-192						
REJ$	#5-74	20-711								
RETURN	9-394	10-427	11-451	12-496	13-543	15-590	20-712	20-722	20-733	21-748
	22-762	23-782	24-790	24-796	24-804	24-812	25-849	26-875	27-890	27-903
SLTDF$	#5-62	5-64								
SOB	20-709	25-841								
STATE$	#5-62	16-603	#16-608	#16-612	#16-619	17-628	#17-630	18-638	#18-640	#18-642
	#18-644	#18-646	#18-648	#18-650	#18-652	#18-655	#18-658	#18-660	#18-663	#18-666
	#18-669	#18-672	#18-675	19-683	#19-685	#19-688				
TR	#5-62	#16-604	#16-605	#16-606	#16-607	#16-609	#16-610	#16-611	#16-613	#16-614
	#16-615	#16-620	#16-621	#17-629	#17-631	#18-639	#18-641	#18-643	#18-645	#18-647
	#18-649	#18-651	#18-653	#18-654	#18-656	#18-657	#18-659	#18-661	#18-662	#18-664
	#18-665	#18-667	#18-668	#18-670	#18-671	#18-673	#18-674	#18-676	#19-684	#19-686

FILEID**NTLCF2

```
                                        .SBTTL  READ - READ NEXT RECORD FROM CETAB

                                ;+
                                ; READ - READ NEXT RECORD FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       CFGSZ=RECORD LENGTH
                                ;
                                ;-
000552                          READ:   GCL$    ,#CFGBF,#132.   ; READ NEXT RECORD
000572  016700  000000G                 MOV     $CLIOS,R0       ; GET GCL IOSB ADDRESS
000576  016067  000002  002444'         MOV     2(R0),CFGSZ     ; SET RECORD LENGTH
000604  105710                          TSTB    (R0)            ; SUCCESS? (CLEARS C-BIT)
000606  100401                          BMI     101$            ; IF MI, NO
000610                          10$:    RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000612  122710  000000G         101$:   CMPB    #IE.EOF,(R0)    ; IS IT END OF FILE?
000616  000261                          SEC                     ; ASSUME THAT IT IS
000620  001773                          BEQ     10$             ; IF EQ, YES - RETURN WITH C-BIT SET
000622  122710  000000G                 CMPB    #IE.RBG,(R0)    ; IS IT BUFFER TOO SMALL?
000626  001403                          BEQ     111$            ; IF EQ, YES
000630                                  EMSG$R  1Q              ; READ FAILURE
000636                          111$:   EMSG$R  1R              ; READ OVERFLOW
```

```
                                        .SBTTL  TPARS STATE TABLES - BUF$DF

                                ;
                                ; BUFFER DEFINITION (BUF$DF)
                                ;
001252                                  STATE$  BUFDF
001252                                   TRAN$   %BUF$DF%,,B.STR,1,SYNERR
001252                                  STATE$  BUFDF1
001252                                   TRAN$   $NUMBR,BUFDF2,B.NUM
001252                                   TRAN$   <',>,BUFDF1,B.ZERO
001252                                   TRAN$   !END,$EXIT,B.ZEND
001252                                  STATE$  BUFDF2
001252                                   TRAN$   <',>,BUFDF1
001252                                   TRAN$   !END,$EXIT,B.END

                                ;
                                ; CHECK FOR END OF SOURCE LINE
                                ;
001252                                  STATE$  END
001252                                   TRAN$   <';>,$EXIT
001252                                   TRAN$   $EOS,$EXIT
```

F 11

```
NTLCF2 - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:24  Page 32
IPARS STATE TABLES -

   1306
   1307 001252                                  STATE$
   1308
   1309          000001                         .END
```

G 11

```
NTLCF2 - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:24  Page 32-1
```

FILEID**NTLCFP

```
NN       NN   TTTTTTTTTT   LL             CCCCCCCC   FFFFFFFFFF   PPPPPPPP
NN       NN   TTTTTTTTTT   LL             CCCCCCCC   FFFFFFFFFF   PPPPPPPP
NN       NN       TT       LL           CC           FF           PP      PP
NN       NN       TT       LL           CC           FF           PP      PP
NNNN     NN       TT       LL           CC           FF           PP      PP
NNNN     NN       TT       LL           CC           FF           PP      PP
NN  NN   NN       TT       LL           CC           FFFFFFFF     PPPPPPPP
NN  NN   NN       TT       LL           CC           FFFFFFFF     PPPPPPPP
NN    NNNN       TT       LL           CC           FF           PP
NN    NNNN       TT       LL           CC           FF           PP
NN       NN       TT       LL           CC           FF           PP              ....
NN       NN       TT       LL           CC           FF           PP              ....
NN       NN       TT       LLLLLLLLLL     CCCCCCCC   FF           PP              ....
NN       NN       TT       LLLLLLLLLL     CCCCCCCC   FF           PP              ....


LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS                 TT
LL           SS                 TT
LL           SS                 TT
LL           SS                 TT
LL             SSSSSS           TT
LL             SSSSSS           TT
LL                   SS         TT
LL                   SS         TT
LL                   SS         TT
LL                   SS         TT
LLLLLLLLLL   SSSSSSSS           TT
LLLLLLLLLL   SSSSSSSS           TT
```

```
                                        .SBTTL  CFLIN - ECHO LAST RECORD READ
                                        .SBTTL  FNLIN - ECHO FILE NAME & LAST RECORD READ
                                ;+
                                ; CFLIN - ECHO THE LAST RECORD READ FROM THE CONFIG FILE
                                ; FNLIN - ECHO FILE NAME AND LAST RECORD READ FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000500                          FNLIN:: CALL    $FNOUT          ; PRINT FILE NAME
000504  012700  000130'         CFLIN:: MOV     #CFGBF,R0       ; GET RECORD BUFFER ADDRESS
000510  016701  000336'                 MOV     CFGSZ,R1        ; AND RECORD SIZE
000514                                  CALLR   $TIOUT          ; PRINT IT ON TI:
```

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
$CLDEQ	= ******	GX	9-303	12-390		
$CLIOS	= ******	GX	10-330	11-352		
$CLQUE	= ******	GX	10-329			
$CLSIZ	= ******	GX	*11-351			
$ERRM4	001234	RG	#8-209	9-310		
$ERRM5	001264	RG	#8-210	9-311		
$ERRM6	001314	RG	#8-211	9-312		
$ERR00	000734	RG	#8-203			
$ERR01	000770	RG	#8-204			
$ERR02	001044	RG	#8-205			
$ERR05	001120	RG	#8-206			
$ERR06	001154	RG	#8-207	14-467		
$ERR09	001174	RG	#8-208			
$ERRP7	000610	RG	#8-200			
$ERRP8	000644	RG	#8-201			
$ERRP9	000700	RG	#8-202			
$ERR1P	001344	RG	#8-212	10-337		
$ERR1Q	001402	RG	#8-213	11-366		
$ERR1R	001440	RG	#8-214	11-367		
$ERR1T	001466	RG	#8-215			
$FLAGS	= ******	GX	9-246			
$FNOUT	= ******	GX	8-212	8-213	13-433	
$GCLEP	= ******	GX	11-351			
$LINKX	= ******	GX	16-557			
$NTCF3	000000	RG	#9-246			
$QCHN	= ******	GX	9-280			
$QCUG	= ******	GX	9-284			
$QDSA	= ******	GX	9-292			
$QDSC	= ******	GX	9-296			
$QDST	= ******	GX	9-269			
$QDTE	= ******	GX	9-267			
$QPSN	= ******	GX	9-253			
$QPVC	= ******	GX	9-282			
$QRDT	= ******	GX	9-271			
$QX29	= ******	GX	9-273			
$QX3P	= ******	GX	9-257			
$REP.N	= ******	GX	8-212	8-213		
$SAVAL	= ******	GX	12-385			
$TIOUT	= ******	GX	12-410	13-436		
.PCHAR	= ******	GX	18-607	18-609		
.PNUMB	= ******	GX	14-455	14-462	14-474	14-488
.PNUMH	= ******	GX	14-453	14-460	14-472	14-486
.PSTCN	= ******	GX	14-444			
.PSTPT	= ******	GX	14-446			

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 6

```
                                        .SBTTL  MACRO DEFINITIONS

                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  EMSG$,EMSG$R,NTLER$,SAVRG,RESRG
```

```
                                        SUB     SKS5,R2         ; RESTORE POINTER TO SCB START
                                        .ENDC
001112  066702  000000G                 ADD     SLHD,R2         ; GET I/O PACKET LISTHEAD
001116  005012                          CLR     (R2)            ;       ...
001120  010262  000002                  MOV     R2,2(R2)        ;       ...
001124  005303                          DEC     R3              ; DONE?
001126  001407                          BEQ     20$             ; IF EQ, YES
001130  010102                          MOV     R1,R2           ; COPY UCB ADDRESS
001132  066002  000000G                 ADD     D.UCBL(R0),R2   ; ADD UCB LENGTH FROM DCB
001136                                  CALL    VAD             ; VALIDATE NEXT UCB ADDRESS
001142  103427                          BCS     131$            ; IF CS, ILLEGAL
001144  000742                          BR      10$             ; SETUP NEXT UCB/SCB PAIR

                                        .IF NDF R$$MPL
001146                          20$:    SWSTK$  50$             ;* ENTER KERNEL MODE 11M

                                        .IFF
                                20$:    SWSTK$  60$             ;* ENTER KERNEL MODE M+

                                        .ENDC

001152  016702  000000G                 MOV     DEVHD,R2        ;* GET DEVICE POINTER
001156  010201                  30$:    MOV     R2,R1           ;* SAVE PREVIOUS DCB ADDRESS
001160  011202                          MOV     (R2),R2         ;* GET NEXT DEVICE IN LIST
001162  001375                          BNE     30$             ;* IF NOT END OF LIST - BRANCH
001164  010011                  40$:    MOV     R0,(R1)         ;* LINK OUR DCB INTO THE LIST
001166  010210                          MOV     R2,(R0)         ;*      ...

001170                          50$:    RETURN                  ;* BACK TO USER MODE

                                        .IF DF  R$$MPL
                                60$:    CALL    $HRCIN          ;* REINIT HRC
                                        RETURN
                                        .ENDC


                                ;
                                ; ERROR CONDITIONS
                                ;
001172  116777  000000G 000000G 101$:   MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
001200                                  EMSG$R  2M              ; READ FAILURE
001206                          111$:   EMSG$R  2Q              ; D.UNIT > D.UNIT+1
001214                          121$:   EMSG$R  2R              ; D.UCB FIELD OUTSIDE TABLE LIMITS
001222  006202                  131$:   ASR     R2              ; ODD ADDRESS?
001224  103003                          BCC     132$            ; IF CC, NO
001226                                  EMSG$R  2S              ; UCB ADDRESS ODD
001234                          132$:   EMSG$R  2T              ; UCB ADDRESS OUTSIDE TABLE LIMITS
001242                          141$:   EMSG$R  2U              ; U.SCB FIELD OUTSIDE TABLE LIMITS
001250  006202                  151$:   ASR     R2              ; ODD ADDRESS?
001252  103003                          BCC     152$            ; IF CC, NO
001254                                  EMSG$R  2V              ; SCB ADDRESS ODD
001262                          152$:   EMSG$R  2W              ; SCB ADDRESS OUTSIDE TABLE LIMITS
```

```
                        .TITLE  MAPX
                        .IDENT  /V05.00/
                        .ENABLE LC
                ;
                ; COPYRIGHT (C) 1982, 1983, 1985  BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A
                ; SINGLE COMPUTER SYSTEM AND  MAY  BE  COPIED  ONLY  WITH  THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE,  OR
                ; ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED  OR  OTHERWISE
                ; MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR  USE  ON  SUCH
                ; SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE  TERMS.  TITLE
                ; TO  AND  OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN
                ; IN DEC.
                ;
                ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
                ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                ; EQUIPMENT CORPORATION.
                ;
                ; DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY  OF
                ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
                ;
                ;
                ; MAPX - Extended Memory Mapping Routines
                ;
                ; IDENT HISTORY:
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
                ;***
                ; Library Macros
                ;***
                        .MCALL  SAVRG,RESRG
                ;***
                ; Global Data
                ;***
000000                  .PSECT  DATA,D
                ;
                ;       Mapping register address
                ;
000000          APR::   .BLKW   1
                ;
                ;       Mapping register contents
                ;
000002          APRSV:: .BLKW   1
```

NTL CREATED BY MACRO ON 29-JUN-85 AT 00:22 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
A.LULU	= 000002		5-64			
BUFFER	000472	R	5-64	#5-64	5-64	
BUFLEN	= 000001		#5-64			
CL.AST	= 004000		#5-64			
CL.ATT	= 000200		5-64	5-64	#5-64	
CL.CAS	= 020000		#5-64			
CL.CMT	= 000001		5-64	5-64	#5-64	
CL.EIF	= 000040		#5-64			
CL.EXN	= 040000		#5-64			
CL.EXT	= 000010		5-64	5-64	#5-64	
CL.IND	= 000002		5-64	5-64	#5-64	
CL.LCC	= 000020		5-64	#5-64		
CL.NPR	= 000400		#5-64			
CL.OPN	= 000100	G	5-64	#5-64	5-70	
CL.RNE	= 000004		#5-64			
CL.XIT	= 100000		#5-64			
C.DSDS	= 000006		5-64			
C.SIZE	= 000054		5-64			
DEFFLG	= 000233		#5-64	5-64		
DFLTNB	000426	R	5-64	#5-64		
EF	= 000001		#5-64	5-64	5-64	5-64
F.DFNB	= 000046		5-64			
F.DSPT	= 000044		5-64			
F.LUN	= 000042		5-64			
F.URBD	= 000020		5-64	5-64	5-64	5-64
G.MCRB	= 000002	G	5-64	5-64	5-64	
N.DVNM	= 000032		5-64			
N.FNAM	= 000006		5-64			
N.FTYP	= 000014		5-64			
N.UNIT	= 000034		5-64			
Q.IOAE	= 000012	G	5-64			
Q.IOFN	= 000002	G	5-64	5-64		
Q.IOPL	= 000014	G	5-64			
Q.IOSB	= 000010	G	5-64	5-64		
S.BFHD	= 000020		5-65	5-65		
S.ETEF	= 000002		5-64			
S.FDB	= 000140		5-64	5-64		
S.FNAM	= 000006		5-64	5-64		
S.FNB	= 000036		5-64			
S.FTYP	= 000002		5-64	5-64		
W.TSEF	= 000002		5-64			
$ASTLV	000210	RG	#5-64			
$CLASN	000162	RG	#5-64			
$CLAST	000172	RG	#5-64			
$CLBUF	= 000310	RG	#5-64			
$CLCSI	000212	RG	#5-64	5-64		
$CLEVL	000206	RG	#5-64			
$CLFDB	000266	RG	#5-64	5-64	5-64	
$CLFLG	000464	RG	#5-64			
$CLIOS	= 000132	RG	#5-64			
$CLMCR	000000	RG	#5-64	5-64		
$CLPMT	000466	RG	#5-64			

NTL CREATED BY MACRO ON 29-JUN-85 AT 00:22 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

```
                                        .SBTTL  CESYM - LOOK FOR SPECIAL SYMBOLS

                                ;+
                                ; CESYM - LOOK FOR SPECIAL COMM EXEC SYMBOLS
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       $RSXNM - IF 0 THEN $SYSNM IS NOT DEFINED (BASE LEVEL PRIOR TO BL28)
                                ;                ELSE ADDRESS OF SYSTEM NAME ($SYSNM)
                                ;       R0,R1,R2=DESTROYED
                                ;
                                ;-

000324  005067  0000006     CESYM:  CLR     $RSXNM          ; ASSUME RSX SYSTEM NAME MATCHES NODE NAME
000330  012700  0000006             MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE SPEC
000334  012701  000006'             MOV     #CEXNAM,R1      ; FOR RSX11M.STB
000340  012702  000020'             MOV     #STBEXT,R2      ;       ...
000344                              CALL    $AZFS           ;       ...

                                    .IF NDF R$$MPL
000350  012700  000040'             MOV     #CBLK1,R0       ; GET SYMBOL DEFINITIONS
                                    .IFF
                                    MOV     #XBLK1,R0       ; GET SYMBOL DEFINITIONS

                                    .ENDC

000354                              CALL    $SYMBL          ; LOOK FOR SYMBOLS IN STB FILE
000360  103471                      BCS     STBERR          ; IF CS, FILE ERROR
000362  012700  0000006             MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE SPEC
000366  012701  000006'             MOV     #CEXNAM,R1      ; FOR [130,5X]CEX.STB
000372  012702  000020'             MOV     #STBEXT,R2      ;       ...
000376                              CALL    $AZFS           ;       ...
000402  012700  000214'             MOV     #IBLK1,R0       ; GET FIRST LOOKUP CONTROL BLOCK ADDRESS
000406                              CALL    $SYMBL          ; LOOK FOR SYMBOLS IN STB FILE
000412  103454                      BCS     STBERR          ; IF CS, FILE ERROR
000414  012700  0000006             MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE SPEC
000420  012701  000012'             MOV     #CETNAM,R1      ; FOR [XXX,5X]CETAB.STB
000424  012702  000020'             MOV     #STBEXT,R2      ;       ...
000430                              CALL    $AZFS           ;       ...
000434  012700  000252'             MOV     #.PDVTB,R0      ; GET ADDRESS OF SYMBOL BLOCK
000440                              CALL    $SYMBL          ; LOOK FOR SYMBOLS IN STB FILE
000444  103437                      BCS     STBERR          ; IF CS, FILE ERROR

                                    .IF DF  R$$MPL

                                    MOV     #10,@STMFC      ; FUNCTION CODE TO DISPATCH 100 MSEC TIMEOUTS
                                    MOV     #1010,@LTMFC    ; FUNCTION CODE TO DISPATCH LONG TIMOUTS
                                    CLR     @CCBAF          ; INITIALIZE ALLOCATION FAILURES
                                    CLR     @RDBAF          ;       ...
                                    CLR     @LDBAF          ;       ...
                                    CLR     @SDBAF          ;       ...
                                    CLR     @CCBAL          ; INITIALIZE NUMBER OF DYNAMIC CCB ALLOCATIONS
                                    MOV     CMFRK,R0        ; GET $CMFRK ADDRESS
```

```
                                        .SBTTL  GTBUF - GET NEXT BUFFER ADDRESS

                                ;+
                                ;
                                ; GTBUF - RETURN ADDRESS OF NEXT BUFFER (RDB/SDB) IN NETWORK POOL
                                ;
                                ; INPUTS:
                                ;       R3      - NUMBER OF BYTES NEEDED ( EVEN )
                                ;       R4      - OLD BIAS
                                ;       R5      - OLD VIRTUAL ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R4      - NEW BIAS
                                ;       R5      - NEW VIRTUAL ADDRESS
                                ;       OTHER REGISTERS UNCHANGED
                                ;
                                ;-

002250                          GTBUF:
002250  010046                          MOV     R0,-(SP)                ; SAVE R0
002252  060305                          ADD     R3,R5                   ; ADD BUFFER SIZE TO VIRTUAL ADDRESS
002254  010500                          MOV     R5,R0                   ; SAVE NUMBER
002256  042705  177700                  BIC     #^C<77>,R5              ; GET NEW VIRTUAL ADDRESS (DIB ONLY)
002262  052705  140000                  BIS     #.BASEB,R5              ; FORCE PAR3/6 BIAS
002266  042700  160000                  BIC     #160000,R0              ; CLEAR MAPPING BIAS BITS
002272                                  ASR$    6,R0                    ; # OF FULL 32-WORD BLOCKS IN BUFFER
002306  060004                          ADD     R0,R4                   ; NEW BIAS
002310  012600                          MOV     (SP)+,R0                ; RESTORE R0

002312                                  RETURN
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 8

MACRO CROSS REFERENCE CREF 04.00

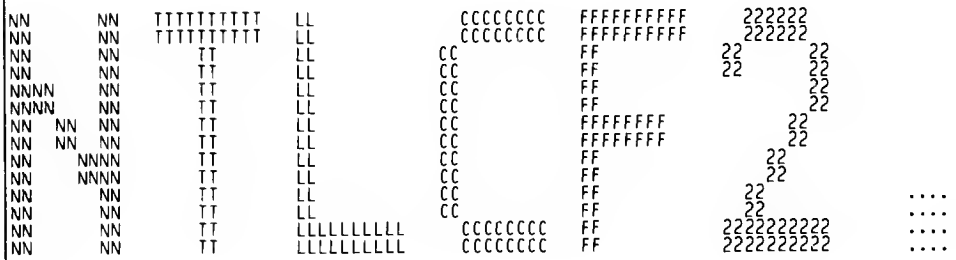
MACRO NAME	REFERENCES									
ASL$	#5-59	7-333	13-755							
ASR$	#5-59	17-985								
BIAS5	#5-59	#5-92	18-1062							
CALL	7-303	7-310	7-311	7-314	7-316	7-322	7-349	7-350	7-353	7-354
	7-356	7-358	8-402	8-411	8-416	8-418	8-423	8-425	9-505	9-506
	10-547	10-552	11-600	11-660	13-737	13-753	13-757	13-765	13-768	13-782
	13-788	13-790	13-796	13-797	14-832	14-863	15-889	15-892	15-908	16-936
	16-939	16-955	18-1011	18-1012	18-1055					
CALLR	7-378	8-484								
CCBDF$	#5-59	5-73								
CRASH	13-803									
CTRDF$	#5-59	5-62								
DHBDF$	#5-59	5-75								
EMSG$R	#5-59	7-376	8-481	8-482	11-666	14-865				
MAP	#5-60	#5-88	18-1060							
NHWDF$	#5-66	5-68								
OPTDF$	#5-59	5-74								
RESMAP	#5-84	18-1076								
RESRG	#5-60	15-907	16-953							
RETC	#5-59	18-1014								
RETURN	7-318	7-371	8-474	9-522	9-528	10-580	11-661	13-786	13-799	14-846
	15-899	15-911	16-946	16-958	17-989	18-1018	18-1077	18-1084		
SAVMAP	#5-80	18-1057								
SAVRG	#5-60	15-903	16-949							
SLTDF$	#5-60	5-63								
SOB	7-370	11-635	11-640	11-650	18-1068					
SWSTK$	7-314	9-505	10-547	13-737	15-889	16-936	18-1011	18-1055		
XPDDB$	#5-60	5-64								

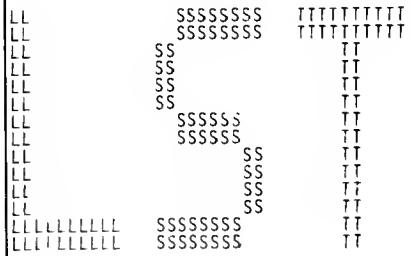
FILEID**NTLCFG

```
                                        .SBTTL  OPEN - OPEN THE CONFIGURATION FILE

                                ;+
                                ; OPEN - OPEN THE CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000772  012700  000000G         OPEN:   MOV     #$CFUIC,R0      ; SETUP ASCIZ FILE SPEC
000776  012701  000000'                 MOV     #CFGNAM,R1      ; FOR [130,10]CETAB.MAC
001002  012702  000006'                 MOV     #CFGEXT,R2      ;       ...
001006                                  CALL    $AZFS           ;       ...
001012                                  CALL    $CLQUE          ; OPEN THE CONFIG FILE
001016  105777  000000G                 TSTB    @$CLIOS         ; SUCCESS? (CLEARS C-BIT)
001022  100401                          BMI     101$            ; IF MI, NO
001024                                  RETURN

                                ;
                                ; ERROR CONDITION
                                ;
001026                          101$:   EMSG$R  1P              ; OPEN FAILURE
```

```
                                        .SBTTL  GETCSR
                                ;
                                ; RETURN NUMERIC CSR VALUE IN R0
                                ;
001622  005767  000000G         GETCSR::TST     .PNUMH          ; IS IT A SINGLE PRECISION NUMBER ?
001626  001404                          BEQ     10$             ; .. YES
001630  026727  000000G 000003          CMP     .PNUMH,#3       ; DOES THE EXCESS = 600000 ?
001636  001010                          BNE     20$             ; .. NO
001640  016700  000000G         10$:    MOV     .PNUMB,R0       ; GET THE LOW ORDER NUMBER
001644  032700  000003                  BIT     #3,R0           ; IS IT AN EVEN MULTIPLE OF 4 ?
001650  001003                          BNE     20$             ; .. NO
001652  020027  160000                  CMP     R0,#160000      ; IS IT AN I/O PAGE ADDRESS ?
001656  103001                          BHIS    30$             ; .. YES (C-BIT IS CLEAR)
001660  000261                  20$:    SEC                     ; ILLEGAL VALUE
001662                          30$:    RETURN
```

FILEID**NTLCF2





NTLCF2 - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:24

Table of contents

```
                                        .SBTTL  Q*** - LOOK FOR ***$DF MACRO
                                ;+
                                ; Q*** - LOOK FOR ***$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4,R5=DESTROYED
                                ;
                                ;-
                                        .ENABL  LSB

000644  012705  000574'         QBUF:   MOV     #BUFDF,R5       ; STATE TABLE ADDRESS
000650  000405                          BR      10$
000652  012705  000654'         QPAR:   MOV     #PARDF,R5       ; STATE TABLE ADDRESS
000656  000402                          BR      10$

000660  012705  001026'         QMFL:   MOV     #MFLDF,R5       ; STATE TABLE ADDRESS
                                ;
000664  005001                  10$:    CLR     R1              ; FULL KEYWORD MATCH LENGTH
000666  012702  000000'                  MOV     #CFGKW,R2       ; KEYWORD TABLE ADDRESS
000672  016703  002444'                  MOV     CFGSZ,R3        ; RECORD LENGTH
000676  012704  002236'                  MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
000702  005067  002452'                  CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
000706                                  CALL    .TPARS          ; GO DO THE PARSE
000712  103003                          BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
000714  005367  002452'                  DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
000720  001401                          BEQ     101$            ; IF EQ, YES
000722                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000724                          101$:   EMSG$R  1T              ; SYNTAX ERROR

                                        .DSABL  LSB
```

```
                                        .SBTTL  TPARS STATE TABLES - PAR$DF
                                ;
                                ; PARTITION DEFINITIONS (PAR$DF)
                                ;
                                STATE$  PARDF
                                 TRAN$   %PAR$DF%,,,1,SYNERR
                                STATE$                  ; FIRST PARTITION NAME
                                .IF NDF M$$MGE
                                 TRAN$   $RAD50,$EXIT,P.PAR1
                                .IFF
                                 TRAN$   <',>,PARDF1
                                 TRAN$   $RAD50,,P.PAR1
                                STATE$
                                 TRAN$   !END,$EXIT,P.END
                                 TRAN$   <',>
                                STATE$  PARDF1          ; SECOND PARTITION NAME
                                 TRAN$   <',>,PARDF2
                                 TRAN$   $RAD50,,P.PAR2
                                STATE$
                                 TRAN$   !END,$EXIT,P.END
                                 TRAN$   <',>
                                STATE$  PARDF2          ; FUDGE FACTOR
                                 TRAN$   <',>,PARDF3
                                 TRAN$   $NUMBR,,P.FUDG
                                STATE$
                                 TRAN$   !END,$EXIT,P.END
                                 TRAN$   <',>
                                STATE$  PARDF3          ; TOP-LOADING FLAG
                                 TRAN$   <',>,PARDF4
                                 TRAN$   "TOP",,,LS.TOP,$FUNC
                                STATE$
                                 TRAN$   !END,$EXIT,P.END
                                 TRAN$   <',>
                                STATE$  PARDF4          ; UMR FLAG OR SIZE
                                 TRAN$   $NUMBR,PARDF5,P.UMR
                                 TRAN$   "NONE",,P.NONE
                                STATE$  PARDF5
                                 TRAN$   !END,$EXIT,P.END
                                 TRAN$   <',>
                                STATE$                  ; SIZE OF SINGLE WORD ALLOCATION POOL
                                 TRAN$   $NUMBR,$EXIT,P.ALL
                                .ENDC
```

Symbol table

```
A         000240R     003 D$DELW  000046          FILDF   000000R     003 LF.PAC= 000200          LS.PRD= 000002
ADDDF     001630R     003 D$END = 000104          FILE  = ****** GX       LF.RDY= 040000          LS.PWF= 040000
A$$CHK= 000000              D$FNB   000034          FILFLG  002152RG    002 LF.REA= 010000          LS.TOP= 002000
A$$CPS= 000000              D$HIOR  000024          FILSPC  000116R     003 LF.SER= 000040          LS.UNF= 020000
A$$PRI= 000000              D$HOST  000022          FLSZ  = 000020 G        LF.TIM= 000010          LS.X25= 000012
A$$TRP= 000000              D$INAC  000044          FL.NTI= ****** GX       LF.UNL= 020000          LS.11D= 004000
B         000260R     003 D$INCT  000042          FMT8    001726RG    002 LF.X2P= 000000          LX.CEX= 000004
BRADJ     001524R     003 D$IPL   000051          FMT8B   001746RG    002 LN.CLD= 000000          LX.LIN= 000006
BRADJ1    001536R     003 D$LID   000020          FMT9    001710RG    002 LN.DUM= 000005          LX.PRO= 000005
BUFDF     000574R     003 D$LNAM  000006          FM.8  = 000000              LN.LDA= 000004          L$$ASG= 000000
BUFDF1    000604R     003 D$LNUM  000014          FM.8B = 000000              LN.LDD= 000003          L$$DRV= 000000
BUFDF2    000630R     003 D$LST   000047          FM.9  = 000000              LN.DAU= 000003          L$$P11= 000001
B.END = ****** GX             D$MAXC  000064          FNBEXT= 000006 G        LN.DFF= 000001          L$$11R= 000000
B.NUM = ****** GX             D$MAXH  000066          FNBLN   002164RG    002 LN.ON = 000000          L.COST  000015
B.STR = ****** GX             D$MAXV  000070          FNBSZ = 000044 G        LN.ODP= 000004          L.CTL   000012
B.ZEND= ****** GX             D$MLL   000040          FNBUFF  002166RG    002 LN.OPE= 000001          L.CVA   177776
B.ZERO= ****** GX             D$MNOD  000041          FNLIN   001046RG        LN.REF= 000002          L.DDM   000002
C         000276R     003 D$NA    000062          F$$LVL= 000001              LN.SER= 000002          L.DDS   000004
CERR      000742RG            D$NBEA  000056          F.NXT   000000              LN.STA= 000017          L.DLC   000003
CFERR     000752RG            D$NBRA  000054          F.USE   000002              LN.SUB= 000360          L.DLM   000006
CFGBF     002236RG    002 D$NEND= 000054          F.VAR   000003              LN.TRI= 000006          L.DLS   000010
CFGEXT    000006R     002 D$NLN   000030          G       000456R     003 LOA     000044R     003 L.FLG   000000
CFGKW     000000RG    004 D$NN    000060          GETSTR= ****** GX       LOGST   002110R     003 L.KRBA  000016
CFGNAM    000000R     002 D$OUTT  000043          GRPUIC= ****** GX       LO.CON  000006          L.LEN = 000022
CFGST     000000RG    003 D$RETF  000050          G$$TPP= 000000              LO.CSR  000006          L.LGST= ****** GX
CFGSZ     002444RG    002 D$RNN   000002          G$$TSS= 000000              LO.CTL  000014          L.MPF   000022
CFLIN     001052RG            D$RTMR  000076          G$$TTK= 000000              LD.LIN  000004          L.NMST  000020
CIRMUL    002056RG    002 D$SEG   000036          G$$WRD= 000000              LO.PAR  000010          L.NSTA  000014
CIRNAM    002050RG    002 D$SER   000032          H       000502R     003 LO.PRI  000010          L.OWNR  000021
CIRNUM    002054RG    002 D$SQRL  000052          HEXSTR  002064RG    002 LO.SCR  000014          L.UNT   000013
CIRUNT    002154RG    002 D$$BUG= 177514          HRDADD= ****** GX       LO.STA  000015          MFLDF   001026R     003
CWARN     000732RG            D$$ISK= 000000          HSAD  = ****** GX       LO.VCT  000012          MRKFL   002454RG    002
C$$CKP= 000000              D$$LI1= 000001          HSAR  = ****** GX       LR.AST  000026          MS.BUF= ****** GX
C$$ORE= 000400              D$$YNC= 000000          HSTARE= ****** GX       LR.CIR  000002          MS.NT1= ****** GX
C$$RSH= 177564              D$$YNM= 000000          HSTNUM= ****** GX       LR.CTL  000004          MS.NT2= ****** GX
D         000316R     003 E       000406R     003 I       000526R     003 LR.EFN  000004          MS.PAR= ****** GX
DADDR = ****** GX             ECLLLN  002460RG    002 IE.EDF= ****** GX       LR.LIN  000002          MULP    000560R     003
DCT   = ****** GX             ECLLNK= ****** GX       IE.RBG= ****** GX       LR.PRM  000030          M$$CRB= 000124
DEACB = ****** GX             ECLNOD  002462RG    002 INAC  = ****** GX       LR.PRO  000002          M$$CRX= 000000
DECDF     001712R     003 END     000644R     003 INCT  = ****** GX       LR.STS  000000          M$$FCS= 000000
DECPT = ****** GX             ENDHEX  002104RG    002 IPL   = ****** GX       LR.TCB  000002          M$$MGE= 000000
DELF  = ****** GX             ENDPAS  002032RG    002 I$$RAR= 000000          LR.UNT  000005          M$$NET= 000000
DELW  = ****** GX             ERRBUF  000022R     002 I$$RDN= 000000          LST   = ****** GX       M$$OVR= 000000
DEV   = ****** GX             ERRMSG  000012R     002 K$$CNT= 177546          LS.AST= 100000          NBRA    002450RG    002
DEVBUF    002142RG    002 EVTDF   002034R     003 K$$CSR= 177546          LS.CEX= 000001          NOAD  = ****** GX
DEVI      000352R     003 EVTMSK  002074R     003 K$$LDC= 000000          LS.CIR= 000010          NDAR  = ****** GX
DEVIC     000362R     003 EXT   = ****** GX       K$$TPS= 000074          LS.CXO= 010000          NDDARE= ****** GX
DEVNM     002060RG    002 EXTEN   000202R     003 LBRA  = 000074          LS.CX5= 000400          NODCT = ****** GX
DIA       000064R     003 E$$XPR= 000000          LD$LP = 000000          LS.DLM= 004000          NDDDF1  001250R     003
DUM       000104R     003 E.CON = ****** GX       LF.ACT= 100000          LS.ECH= 001000          NDDID = ****** GX
DVSZ  = 000010 G              E.FIL = ****** GX       LF.BRD= 000400          LS.FDX= 004000          NODNM = ****** GX
D$AMXC    000072              E.MON = ****** GX       LF.BWT= 000007          LS.HDX= 010000          NODNUM= ****** GX
D$AMXH    000074              E.MSK = ****** GX       LF.ENA= 002000          LS.LIN= 000003          N$$ACC= 000001
D$ANN     000000              E.NUM = ****** GX       LF.LPB= 001000          LS.LMC= 000007          N$$BUF= 000001
D$BRPR    000102              F       000432R     003 LF.MDC= 000100          LS.NTI= 004000          N$$LDV= 000001
D$BRTM    000100              FIELDC= 000003 G        LF.MFL= 004000          LS.ON = 000011          N$$MCP= 000001
D$DELF    000045              FILBUF  002122RG    002 LF.MTP= 000020          LS.OPT= 000400          N$$MLL= 000001
```

NTLCFP - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:29

Table of contents

5-	52	MACRO DEFINITIONS
6-	66	LOCAL SYMBOL DEFINITIONS
7-	82	LOCAL DATA
8-	194	ERROR MESSAGES
9-	231	$NTCF3 - SCAN CETAB.MAC FOR PSI COMPONENTS
10-	314	OPEN - OPEN THE CONFIGURATION FILE
11-	339	READ - READ NEXT RECORD
12-	369	CERR,CFERR - PRINT ERROR MESSAGE
13-	421	CFLIN - ECHO LAST RECORD READ
13-	422	FNLIN - ECHO FILE NAME & LAST RECORD READ
14-	438	COMMON ACTION ROUTINES - LINE-ID
15-	497	PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
16-	534	LNKEND - LINK BLOCK AT END OF LIST
17-	560	FNDPDV - FIND PDV
18-	592	STRNXT - STORE NEXT DIGIT

NTLCFP - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:29 Page 4

```
                                        .SBTTL  COMMON ACTION ROUTINES - LINE-ID

                                        .ENABL  LSB

                                ;
                                ; DEVICE NAME
                                ;
000520  022767  000003  000000G DEVNAM::CMP     #3,.PSICN               ; VALID DEVICE NAME?
000526  103425                        BLO     101$                    ; BR IF NO
000530  016700  000000G               MOV     .PSIPT,R0               ; GET ADDRESS OF FIRST CHARACTER
000534                                 CALL    $CAT5                   ; CONVERT TO RAD50
000540  010167  000346'               MOV     R1,LINNAM               ; SAVE DEVICENAME
000544                                 RETURN
                                ;
                                ; DEVICE CONTROLLER NUMBER
                                ;
000546  005767  000000G         DEVCTL::TST     .PNUMH                  ; LEGAL CONTROLLER NUMBER VALUE?
000552  001013                        BNE     101$                    ; BR IF NO
000554  016767  000000G 000350'       MOV     .PNUMB,LINCTL           ; SAVE CONTROLLER NUMBER
000562                                 RETURN
                                ;
                                ; DEVICE UNIT NUMBER
                                ;
000564  005767  000000G         DEVUNT::TST     .PNUMH                  ; LEGAL UNIT NUMBER VALUE?
000570  001004                        BNE     101$                    ; BR IF NO
000572  116767  000000G 000351'       MOVB    .PNUMB,LINUNT           ; SAVE UNIT NUMBER
000600                                 RETURN
                                ;
                                ; ERRORS
                                ;
000602                          101$:   EMSG$R  06                      ; ILLEGAL LINE-ID
                                        .DSABL  LSB
                                ;
                                ; CHECK FOR LEGAL BLOCK SIZE VALUE
                                ;
000610  005767  000000G         CHKBLK::TST     .PNUMH                  ; DOUBLE WORD VALUE?
000614  001012                        BNE     10$                     ; BR IF YES - ERROR
000616  016700  000000G               MOV     .PNUMB,R0               ; GET BLOCK SIZE VALUE
000622  022700  000040                CMP     #BLKSMN,R0              ; IS BLOCK SIZE VALUE WITHIN RANGE?
000626  101005                        BHI     10$                     ; BR IF NO - ERROR
000630  022700  002000                CMP     #BLKSMX,R0              ; ...
000634  103402                        BLO     10$                     ; BR IF NO - ERROR
000636  000241                        CLC                             ; INDICATE LEGAL BLOCK SIZE VALUE
000640  000401                        BR      20$
000642  000261                10$:    SEC                             ; INDICATE ILLEGAL VALUE
000644                          20$:    RETURN
                                ;
                                ; CHECK FOR LEGAL WINDOW SIZE VALUE
                                ;
000646  005767  000000G         CHKWND::TST     .PNUMH                  ; DOUBLE WORD VALUE?
000652  001010                        BNE     10$                     ; BR IF YES - ERROR
000654  016700  000000G               MOV     .PNUMB,R0               ; GET BLOCK SIZE VALUE
000660  003405                        BLE     10$                     ; BR IF ILLEGAL VALUE
000662  022700  000177                CMP     #WNDSMX,R0              ; IS VALUE IN RANGE?
000666  103402                        BLO     10$                     ; BR IF NO - ERROR
000670  000241                        CLC                             ; INDICATE LEGAL WINDOW SIZE VALUE
000672  000401                        BR      20$
000674  000261                10$:    SEC                             ; INDICATE ILLEGAL VALUE
```

NTLCFP CREATED BY MACRO ON 29-JUN-85 AT 00:29 PAGE 6

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-57	15-524								
CALL	9-250	9-251	9-253	9-255	9-257	9-262	9-267	9-269	9-271	9-273
	9-278	9-280	9-282	9-284	9-290	9-292	9-294	9-296	9-298	9-303
	10-328	10-329	11-351	12-385	12-390	12-397	12-410	12-413	13-433	14-447
	16-552	16-557	18-612							
CALLR	13-436									
EMSG$R	#5-57	9-310	9-311	9-312	10-337	11-366	11-367	14-467		
GCL$	#5-57	11-351								
NTLDF$	#5-57	5-63								
NTLER$	#5-57	8-200	8-201	8-202	8-203	8-204	8-205	8-206	8-207	8-208
	8-209	8-210	8-211	8-212	8-213	8-214	8-215			
RETC	#5-58									
RETURN	9-305	10-332	11-356	12-418	14-449	14-456	14-463	14-482	14-495	15-532
	16-558	17-590	18-616							
SLTDF$	#5-58	5-64								
SOB	17-583									

FILEID**NTLCSR

```
                                        .SBTTL  LOCAL DATA

                                ;
                                ; LOCAL DATA
                                ;

000000                          .PSECT  DATA,D

                                .NLIST  BEX

                                .IF DF  R$$11M
                                ;
                                ; SYMBOL TABLE EXTENSION
                                ;
000000  123  124  102   STBEXT: .ASCIZ  "STB"

                                ;
                                ; DEVICE NAME PRECEDED BY "$"
                                ;
000004  044  130  130   DNAME:  .ASCII  "$XX"

                                .IFTF   ; DF R$$11M
                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
000007  015             ERRMSG: .BYTE   15
000010  116  124  114           .ASCII  "NTL -- "
000017                  ERRBUF: .BLKB   70.

                                .EVEN

                                .IFT    ; DF R$$11M
                                ;
                                ; SYMBOL LOOKUP CONTROL BLOCKS
                                ;
000126  000140' 000000  DBLK1:  .WORD   DBLK2,0         ; THIS MUST BE THE FIRST CONTROL BLOCK
000132  126230  020564          .RAD50  "$XXEND"
000136  000000          LENGTH: .WORD   0
000140  000000  000000  DBLK2:  .WORD   0,0
000144  126230  014572          .RAD50  "$XXDCB"
000150  000000          DCB:    .WORD   0

                                ;
                                ; DCB DISPATCH TABLE NAME (THIS SYMBOL COMES FROM THE PROCESS STB FILE)
                                ;
000152  126230  076534  DUMMY:  .RAD50  "$XXTBL"

                                ;
                                ; STB FILE ERRORS
                                ;
000156  000220'         ETAB:   .WORD   $ERR2B          ; OPEN FAILURE
000160  000256'                 .WORD   $ERR2C          ; READ FAILURE
000162  000314'                 .WORD   $ERR2D          ; READ OVERFLOW
000164  000342'                 .WORD   $ERR2E          ; BAD STB FORMAT

                                ;
                                ; FLAG WHETHER OR NOT DEVICE STRUCTURES ALREADY EXIST
```

```
                                        .SBTTL  RVAD - RELOCATE AND VALIDATE DATA STRUCTURES
                                        .SBTTL  VAD - VALIDATE DATA STRUCTURES


                                ;+
                                ;
                                ; RVAD - RELOCATE AND VALIDATE DATA STRUCTURE ADDRESS
                                ; VAD  - VALIDATE DATA STRUCTURES
                                ;
                                ;-
001270  166702  000170'         RVAD:   SUB     BASE,R2         ; ADJUST BY TKB BASE ADDRESS
001274  066702  000000G                 ADD     $LBUF,R2        ; AND BY BUFFER ADDRESS FROM THE POOL
001300  032702  000001          VAD:    BIT     #1,R2           ; IS ADDRESS ODD?
001304  000261                          SEC                     ; IN CASE IT IS
001306  001005                          BNE     10$             ; IF NE, YES
001310  020267  000000G                 CMP     R2,$LBUF        ; IS ADDRESS < START OF BUFFER?
001314  103402                          BCS     10$             ; IF CS, YES
001316  026702  000174'                 CMP     TABEND,R2       ; SET C-BIT IF ADDRESS > END OF BUFFER
001322                          10$:    RETURN
```

```
MAPX    MACRO V05.03b Sunday 30-Jun-85 00:00  Page 4-1


     58                                 ;
     59 000000                                .PSECT  .MAPX
```



```
MAPX    MACRO V05.03b Sunday 30-Jun-85 00:00  Page 5
$MAPX - MAP TO EXTENDED MEMORY
```

NTL CREATED BY MACRO ON 29-JUN-85 AT 00:22 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
$CLQIO	000122 RG	#5-64	5-64						
$CLRTN	= ****** GX	5-64							
$CLSBK	000176 RG	5-64	#5-6						
$CLSEF	000156 RG	#5-64							
$CLSIZ	= 000306 RG	#5-64							
$CLSRV	000470 RG	#5-64							
$CLTBF	000202 RG	#5-64							
$CLTSZ	000204 RG	#5-64							
$CLWEF	000152 RG	#5-64							
$PSHPT	000174 RG	#5-64							
$$$ARG	= 000003	#5-64	5-64	#5-64	5-64	#5-64	5-64	#5-64	5-64	5-64
$$$GLB	= ******	5-64	5-64	5-64	5-64	5-64				
$$$T1	= 000002	#5-64								
$$$T2	= 000166 R	#5-64	5-64							
.FSRCB	= ****** G	5-65								
...PC1	= 000266 R	#5-64	5-64	5-64	5-64	5-64	5-64	#5-64	5-64	5-64
		5-64	5-64	5-64	#5-64	5-64	5-64			
...PC2	= 000464 R	#5-64	5-64	#5-64	5-64	#5-64	5-64	#5-64	5-64	5-64
		#5-64	5-64	5-64	#5-64	5-64	#5-64	5-64	#5-64	5-64
...PC3	= 000266 R	#5-64	5-64							

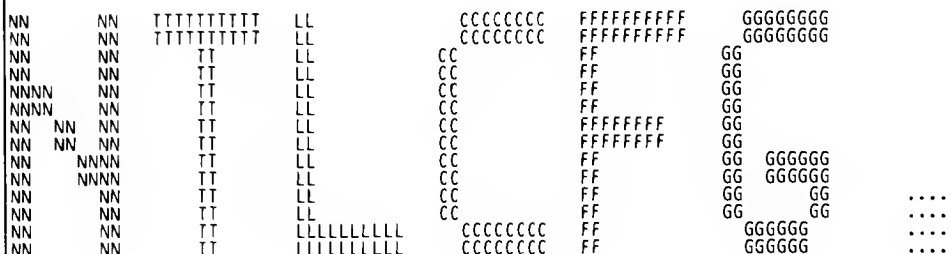
NTL CREATED BY MACRO ON 29-JUN-85 AT 00:22 PAGE 3

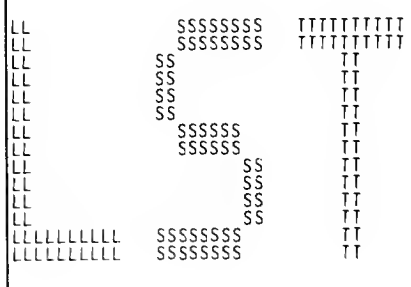
MACRO CROSS REFERENCE CREF 04.00

```
                                        ADD     #4,R0           ; ACCESS $CMFRK+4
                                        MOV     R0,R1           ; SAVE ADDRESS
                                        CLR     (R0)+           ; INITIALIZE FORK BLOCK LISTHEAD
                                        MOV     R1,(R0)         ;       ...
                                        .ENDC
000446  012700  000226'                 MOV     #IBLK2,R0       ; POINT TO SYMBOL BLOCK FOR SYSTEM NAME
000452  005760  000002                  TST     2(R0)           ; IS IT DEFINED?
000456  001406                          BEQ     7$              ; BR IF NO - OK
000460  016767  000000G 000000G         MOV     SYSNM,$RSXNM    ; GET SYMBOL VALUE
000466  105767  000000G                 TSTB    $RSXNM          ; IS DECNET SUPPORTED?
000472  001421                          BEQ     102$            ; BR IF NO
                                        .IF DF  R$$MPL
                                7$:     MOV     #$CXUIC,R0      ; SETUP ASCIZ FILE STRING
                                        MOV     #VEXNAM,R1      ;       ...
                                        MOV     #STBEXT,R2      ;       ...
                                        CALL    $AZFS           ;       ...
                                        MOV     #VXBLK1,R0      ; GET VECTORED SYMBOL BLOCK ADDRRESSES
                                        CALL    $SYMBL          ; DEFINE SYMBOL VALUES
                                        BCS     STBERR          ; IF ERROR - BRANCH
                                        .IFF
000474  012700  000040'         7$:     MOV     #CBLK1,R0       ; START AT FIRST SYMBOL BLOCK
000500  005760  000002          10$:    TST     2(R0)           ; IS SYMBOL DEFINED?
000504  001404                          BEQ     20$             ; IF EQ, NO - DON'T WORRY
000506  026060  000010  000012          CMP     10(R0),12(R0)   ; DOES VALUE MATCH TKB DEFINITION?
000514  001005                          BNE     101$            ; IF NE, NO - VERY BAD
000516  011000                  20$:    MOV     (R0),R0         ; GET NEXT SYMBOL BLOCK ADDRESS
000520  022700  000120'                 CMP     #XBLK1,R0       ; IS IT THE FIRST PROCESS VERIFICATION BLOCK?
000524  001365                          BNE     10$             ; IF NE, NOT AT END OF LIST
000526                                  RETURN
                                        .ENDC
                                ;
                                ; ERROR CONDITIONS
                                ;
000530                          101$:   EMSG$R  48              ; CEX/RSX INCOMPATIBLE
000536                          102$:   EMSG$R  QY              ; DECNET NOT SUPPORTED
000544  016000  000332'         STBERR: MOV     ETABi-2(R0),R0  ; GET ERROR MESSAGE CONTROL BLOCK ADDRESS
000550                                  CALLR   @(R0)+          ; PRINT ERROR MESSAGE
```

```
                                        .SBTTL  $ALTLC - SET UP XPT CIRCUIT COUNTER BLOCK

                                ;+
                                ; $ALTLC - SET UP CIRCUIT COUNTER BLOCK
                                ;
                                ; INPUTS:
                                ;       $$HTIM - HELLO TIMER VALUE
                                ;       $$LTIM - LISTEN TIMER VALUE
                                ;       .NBRA  - BROADCAST ROUTER ADJACENCIES
                                ;       .RPRI  - ROUTER PRIORITY
                                ;
                                ; OUTPUTS:
                                ;       COUNTER BLOCK IS SET UP
                                ;-
002314  012702  114224          $ALTLC::MOV     #^RXPT,R2       ; GET XPT'S PROCESS NAME

002320                                  SWSTK$  30$             ;; ENTER SYSTEM STATE
002324                                  CALL    @PDVID          ;; LOOK FOR XPT'S PDV
002330  103006                          BCC     10$             ;; BR IF FOUND IT
002332                                  RETC    R0              ;; ELSE SET USER C-BIT
002344  000404                          BR      20$
002346  067702  000000G         10$:    ADD     @PDVTA,R2       ;; INDEX INTO PDV VECTOR TABLE
002352  011267  000034'                 MOV     (R2),XPTPDV     ;; GET PDV ADDRESS
002356                          20$:    RETURN

002360  103540                  30$:    BCS     100$            ; BR IF XPT NOT FOUND????
002362  005000                          CLR     R0              ; ASSUME POINT-TO-POINT LINE
002364  016704  000000G                 MOV     $SLTA,R4        ; GET SLT ADDRESS
002370  005001                          CLR     R1              ; ASSUME NOT A MULTIPOINT LINE
002372  032764  000020  000000          BIT     #LF.MTP,L.FLG(R4) ; IS THIS A MULTIPOINT LINE?
002400  001402                          BEQ     33$             ; BR IF NO
002402  116401  000014                  MOVB    L.NSTA(R4),R1   ; GET NUMBER OF STATIONS
002406  017703  000000G         33$:    MOV     @SLTMA,R3       ; POINT TO SLT VECTOR TABLE
002412  005046                          CLR     -(SP)           ; INITIALIZE SYSTEM LINE NUMBER
002414  022304                  35$:    CMP     (R3)+,R4        ; IS THIS THE CORRECT SLT?
002416  001402                          BEQ     36$             ; BR IF YES
002420  005216                          INC     (SP)            ; ELSE UPDATE SYSTEM LINE NUMBER
002422  000774                          BR      35$             ; AND TRY AGAIN
002424  006316                  36$:    ASL     (SP)            ; MAKE IT A WORD INDEX
002426  067716  000000G                 ADD     @LLCTA,(SP)     ; INDEX INTO REVERSE MAPPING TABLE
002432  013604                          MOV     @(SP)+,R4       ; GET CHANNEL NUMBER AND PDV
002434  100004                          BPL     50$             ; BR IF NOT MULTIPOINT
002436  006304                          ASL     R4              ; GET STATION TABLE ADDRESS
002440  010403                          MOV     R4,R3           ; SAVE STATION TABLE ADDRESS
002442  060004                  40$:    ADD     R0,R4           ; POINT TO CORRECT ENTRY IN STATION TABLE
002444  011404                          MOV     (R4),R4         ; GET CHANNEL NUMBER AND PDV
002446  016702  000034'         50$:    MOV     XPTPDV,R2       ; GET XPT'S PDV ADDRESS
002452  062702  000000G                 ADD     #Z.MAP,R2       ; POINT TO FORWARD MAPPING TABLE
002456  005046                          CLR     -(SP)           ; GET CHANNEL NUMBER
002460  110416                          MOVB    R4,(SP)         ; ...
002462  006316                          ASL     (SP)            ; MAKE IT A WORD INDEX
002464  062602                          ADD     (SP)+,R2        ; POINT TO SLN, STA PAIR
002466  011204                          MOV     (R2),R4         ; GET SLN, STA PAIR
002470  016067  000000G 000024'         MOV     $$HTIM(R0),HTIM ; GET HELLO TIMER FOR CURRENT STATION
002476  016067  000000G 000026'         MOV     $$LTIM(R0),LTIM ; GET LISTEN TIMER FOR CURRENT STATION
```

FILEID**NTLCFG


NN NN TTTTTTTTTT LL CCCCCCCC FFFFFFFFFF GGGGGGGG
NN NN TTTTTTTTTT LL CCCCCCCC FFFFFFFFFF GGGGGGGG
NN NN TT LL CC FF GG
NN NN TT LL CC FF GG
NNNN NN TT LL CC FF GG
NNNN NN TT LL CC FF GG
NN NN NN TT LL CC FFFFFFFF GG
NN NN NN TT LL CC FFFFFFFF GG
NN NNNN TT LL CC FF GG GGGGGG
NN NNNN TT LL CC FF GG GGGGGG
NN NN TT LL CC FF GG GG ....
NN NN TT LL CC FF GG GG ....
NN NN TT LLLLLLLLLL CCCCCCCC FF GGGGGG ....
NN NN TT LLLLLLLLLL CCCCCCCC FF GGGGGG ....



NTLCFG - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:23
Table of contents

```
                                        .SBTTL  READ - READ NEXT RECORD

                                ;+
                                ; READ - READ NEXT RECORD FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       CFGSZ=RECORD LENGTH
                                ;-
001034                          READ:   GCL$    ,#CFGBF,#132.   ; READ NEXT RECORD
001054  016700  000000G                 MOV     $CLIOS,R0       ; GET GCL IOSB ADDRESS
001060  016067  000002  000352'         MOV     2(R0),CFGSZ     ; SET RECORD LENGTH
001066  105710                          TSTB    (R0)            ; SUCCESS? (CLEARS C-BIT)
001070  100401                          BMI     101$            ; IF MI, NO
001072                          10$:    RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
001074  122710  000000G         101$:   CMPB    #IE.EOF,(R0)    ; IS IT END OF FILE?
001100  000261                         SEC                     ; ASSUME THAT IT IS
001102  001773                         BEQ     10$             ; IF EQ, YES - RETURN WITH C-BIT SET
001104  122710  000000G                 CMPB    #IE.RBG,(R0)    ; IS IT BUFFER TOO SMALL?
001110  001403                         BEQ     111$            ; IF EQ, YES
001112                                  EMSG$R  1Q              ; READ FAILURE
001120                          111$:   EMSG$R  1R              ; READ OVERFLOW
```

```
                                        .SBTTL  LLC$DF ACTION ROUTINES

                                ;
                                ; FIRST CHARACTER OF DEVICE NAME (LLC$DF)
                                ;
001664  116767  000000G 000000G L.CH1:  MOVB    .PCHAR,$$DEV    ; COPY THE CHARACTER
001672                                  RETURN

                                ;
                                ; SECOND CHARACTER OF DEVICE NAME (LLC$DF)
                                ;
001674  116767  000000G 000001G L.CH2:  MOVB    .PCHAR,$$DEV+1  ; COPY SECOND CHARACTER
001702                                  RETURN

                                ;
                                ; COUNTER TIMER (SLT$DF)
                                ;
001704  005767  000000G         L.CTIM: TST     .PNUMH          ; VALID COUNTER TIMER VALUE?
001710  001004                          BNE     101$            ; BR IF NO
001712  016767  000000G 000000G        MOV     .PNUMB,$CTIM    ; ELSE SAVE COUNTER TIMER VALUE
001720                                  RETURN
001722                          101$:   EMSG$R  YR              ; ILLEGAL COUNTER TIMER VALUE

                                ;
                                ; ECL SEGMENT SIZE
                                ;
001730  017700  000000G         L.SEG:  MOV     @DECPT,R0       ; COPY HOME BLOCK ADDRESS
001734  016760  000000G 000036          MOV     .PNUMB,D$SEG(R0); STORE SEGMENT SIZE
001742                                  RETURN
```

Table of contents

5- 54 MACRO DEFINTIONS
6- 75 LOCAL DATA
7- 100 ERROR MESSAGES
8- 268 $NTICF - PROCESS CETAB FOR MISSING PARAMETERS
9- 418 OPEN - OPEN THE CONFIG FILE
10- 445 READ - READ NEXT RECORD FROM CETAB
11- 477 Q*** - LOOK FOR ***$DF MACRO
12- 518 CERR - PRINT ERROR MESSAGE
12- 519 CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
12- 520 CWARN - PRINT WARNING MESSAGE
13- 574 CFLIN - ECHO THE LAST RECORD READ
13- 575 FNLIN - ECHO FILE NAME AND LAST RECORD READ
14- 594 REP8B - REPLACE PARTITION NAME
14- 595 REP8C - REPLACE REGION NAME
15- 631 REP.NN - REPLACE SIGNED DECIMAL NUMBER
16- 658 REPND - REPLACE NODE NAME
19- 770 TPARS STATE TABLES - SER$DF
21- 899 TPARS STATE TABLES - BUF$DF
22- 921 TPARS STATE TABLES - PAR$DF
23- 964 TPARS STATE TABLES - NOD$DF
24- 1017 TPARS STATE TABLES - LLCDF,DLCDF,DDM$DF
25- 1046 TPARS STATE TABLES - SLT$DF
26- 1105 STADF,ADDDF STATE TABLES
27- 1133 TPARS STATE TABLES - OBJ$DF
29- 1239 TPARS STATE TABLES - REM$DF
30- 1265 TPARS STATE TABLES - EVT$DF
31- 1299 TPARS STATE TABLES -

```
                                        .SBTTL  CERR - PRINT ERROR MESSAGE
                                        .SBTTL  CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
                                        .SBTTL  CWARN - PRINT WARNING MESSAGE

                                ;+
                                ; CERR - PRINT ERROR MESSAGE
                                ; CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
                                ; CWARN - PRINT WARNING MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET IF ERROR MESSAGE
                                ;       RETURN TO $NTICF'S CALLER OR BACK TO CALLER
                                ;
                                ;-
                                        .ENABL  LSB
000732                          CWARN::
000732                                  CALL    $SAVAL          ; SAVE ALL REGISTERS
000736  005046                          CLR     -(SP)           ; WARNING MESSAGE
000740  000405                          BR      5$              ; JOIN COMMON CODE
000742                          CERR::
000742  010046                          MOV     R0,-(SP)        ; SAVE R0
000744                                  CALL    $CLDEQ          ; CLOSE THE CONFIG FILE
000750  012600                          MOV     (SP)+,R0        ; RESTORE R0
000752                          CFERR::
000752  010746                          MOV     PC,-(SP)        ; ERROR MESSAGE
000754  010005                  5$:     MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
000756  012704  000022'                  MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
000762  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
000764                                  CALL    @(R5)+          ; DO THE FORMATTING
000766  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
000770  001403                          BEQ     20$             ; IF EQ, YES
000772  112324                  10$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
000774  001376                          BNE     10$             ;  ...
000776  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001000  012503                  20$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
001002  112524                  30$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
001004  001376                          BNE     30$             ;  ...
001006  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001010  012700  000012'                  MOV     #ERRMSG,R0      ; GET ERROR MESSAGE ADDRESS
001014  010401                          MOV     R4,R1           ; COPY END ADDRESS
001016  160001                          SUB     R0,R1           ; GIVING LENGTH
001020                                  CALL    $TIOUT          ; PRINT IT
001024  005703                          TST     R3              ; SHOULD SECOND LINE BE PRINTED?
001026  001401                          BEQ     40$             ; IF EQ, NO
001030                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
001032  005726                  40$:    TST     (SP)+           ; ERROR OR WARNING MESSAGE ?
001034  001403                          BEQ     RTSPC           ; IF EQ, WARNING
001036  016706  002446'                  MOV     SPSAV,SP        ; RESTORE SP FROM ENTRY
001042  000261                          SEC                     ; RETURN C-BIT SET
001044                          RTSPC:: RETURN
                                        .DSABL  LSB
```

```
                                        .SBTTL  TPARS STATE TABLES - NOD$DF
                                ;
                                ; MARKED-FOR-LOAD/ENABLE SCAN
                                ;
001252                                  STATE$  MFLDF

001252                                   TRAN$   %LLC$DF%,PRODF,,1,SYNERR
001252                                   TRAN$   %DLC$DF%,PRODF,,1,SYNERR
001252                                   TRAN$   %DDM$DF%,PRODF,,1,SYNERR
001252                                   TRAN$   %SLT$DF%,SLTDF,,1,SYNERR
001252                                   TRAN$   %STA$DF%,STADF,,1,SYNERR
001252                                   TRAN$   %ADD$DF%,ADDDF,,1,SYNERR
001252                                   TRAN$   %OBJ$DF%,OBJDF,,1,SYNERR
001252                                   TRAN$   %REM$DF%,REMDF,,1,SYNERR
001252                                   TRAN$   %EVT$DF%,EVTDF,,1,SYNERR
001252                                   TRAN$   %LOG$ST%,LOGST,,1,SYNERR
001252                                   TRAN$   %SER$DF%,SERDF,,1,SYNERR
001252                                   TRAN$   %FIL$DF%,FILDF,,1,SYNERR
001252                                   TRAN$   %NOD$DF%,,,1,SYNERR

                                ;
                                ; NOD$DF
                                ;
001252                                  STATE$
001252                                   TRAN$   LBRA,,NODNM
001252                                  STATE$
001252                                   TRAN$   !NODDF1
001252                                  STATE$
001252                                   TRAN$   <',>
001252                                  STATE$
001252                                   TRAN$   LBRA,,NODID
001252                                  STATE$
001252                                   TRAN$   !NODDF1
001252                                  STATE$
001252                                   TRAN$   <',>
001252                                  STATE$
001252                                   TRAN$  $NUMBR,,NODARE
001252                                  STATE$
001252                                   TRAN$  <',>
001252                                  STATE$
001252                                   TRAN$   $NUMBR,,NODNUM
001252                                  STATE$
001252                                   TRAN$   <',>
001252                                  STATE$
001252                                   TRAN$  $NUMBR,,HSTARE
001252                                  STATE$
001252                                   TRAN$  <',>
001252                                  STATE$
001252                                   TRAN$   $NUMBR,$EXIT,HSTNUM
001252                                  STATE$  NODDF1
001252                                   TRAN$   RBRA,$EXIT,STRCH
001252                                   TRAN$   $ANY,NODDF1
```

Symbol table

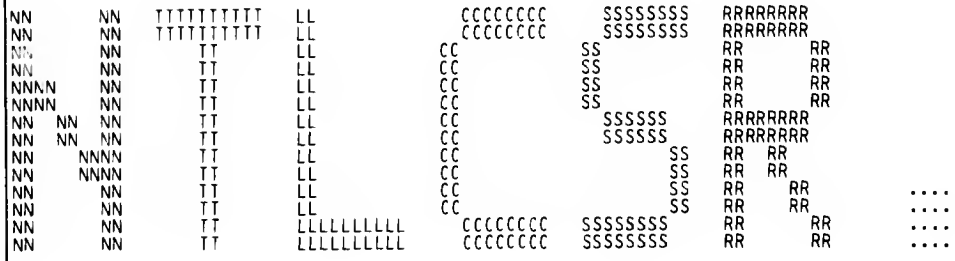
```
N$$MOV= 000010          R$$DER= 000000          SF.PSS= 000040          S.UNT = ****** GX       $ERRXS  001344RG  002
N$$NCT= 000001          R$$K11= 000001          SF.REA= 000010          TER     000024R  003    $ERRXT  001412RG  002
N$$PEM= 000001          R$$SND= 000000          SF.SEC= 000400          TOK     000152R  003    $ERRXU  001470RG  002
OBJDF   001634R  003   R$$11M= 000000          SF.SER= 000001          T$$KMG= 000000          $ERRXV  001054RG  002
OBJ2    001660R  003   R.ARE = ****** GX       SF.SVC= 000002          T$$MIN= 000000          $ERRXW  001604RG  002
OPEN    000510R        R.NAME= ****** GX       SF.TER= 001000          UIC     000132R  003    $ERRXX  001646RG  002
OUTT  = ****** GX      R.NUM = ****** GX       SF.UNL= 000040          UICBUF  002106RG 002    $ERRXY  001536RG  002
O.CPY = ****** GX      SCHEL   001516R  003   SLTDF   001350R  003   UISZ  = 000014 G          $ERR1P  000130RG  002
O.END = ****** GX      SCHELO  001506R  003   SLTDF1  001426R  003   USRUIC= ****** GX        $ERR1Q  000166RG  002
O.FLAG= ****** GX      SCTIM   001474R  003   SLTDF2  001434R  003   USRUI1= ****** GX        $ERR1R  000224RG  002
O.NAME= ****** GX      SERADD  002040RG 002   SLTDF3  001374R  003   V$$CTR= 001000           $ERR1T  000252RG  002
O.NUM = ****** GX      SERBEG  002010RG 002   SLTMA = ****** GX       XPTLLN  002456RG 002    $ERR1Z  000342RG  002
O.VFY = ****** GX      SERCIR  002016RG 002   SLTNM = ****** GX       X$$DBT= 000000          $EXIT = 000000
PARDF   000654R  003   SERCR = ****** GX       SPSAV   002446R  002   X.FLG = ****** GX       $FAIL = 177777
PARDF1  000704R  003   SERCUR  002156RG 002   STADF   001544R  003   ZF.MFL= ****** GX       $FLAGS= ****** GX
PARDF2  000726R  003   SERDCT  002036RG 002   STADF1  001562R  003   ZSS     000564R  003    $FNOUT= ****** GX
PARDF3  000750R  003   SERDEV  002020RG 002   STAN    002002RG 002   Z.FLG = ****** GX        $FUNC = ****** GX
PARDF4  000774R  003   SERDF   000214R  003   STRADD  002232RG 002   $ALPHA= 000022           $GCLEP= ****** GX
PARDF5  001006R  003   SERDMP  002032RG 002   STRCH = ****** GX       $ANY  = 000020          $LAMDA= 000000
PAS   = ****** GX      SERDV = ****** GX       STRPT   002234RG 002   $AZFS = ****** GX       $LSTAD  002006RG  002
PDVNM = ****** GX      SEREND= ****** GX       STRSIZ  002104RG 002   $BLANK= 000006          $LSTRM  002004RG  002
PDVTA = ****** GX      SERFIN  002166RG 002   SYNERR  002452RG 002   $CBDSG= ****** GX       $MISS = ****** GX
PRMFL = ****** GX      SERFLG  002010RG 002   S$$WRG= 000000          $CFUIC= ****** GX       $NTICF  000000RG
PRODF   001262R  003   SERHST  002014RG 002   S$$YSZ= 007600          $CLBUF= ****** GX       $NUMBR= 000002
PRODF1  001270R  003   SERINI= ****** GX       S.ADD   000004          $CLDEQ= ****** GX       $QBRA = ****** GX
PRODF2  001320R  003   SERMUL= ****** GX       S.BRA = ****** GX       $CLIOS= ****** GX       $QDEC = ****** GX
PRONM = ****** GX      SERMUX= ****** GX       S.CIR   000016          $CLQUE= ****** GX       $QMLMC= ****** GX
P$$P45= 000000         SERNOD  002012RG 002   S.COST  000001          $CLSIZ= ****** GX       $QXPT = ****** GX
P$$WRD= 000000         SEROFF  002160RG 002   S.CTIM= ****** GX       $C5TA = ****** GX       $RAD50= 000016
P.ALL = ****** GX      SERPAS  002022RG 002   S.CTL = ****** GX       $DIGIT= 000024          $REP.N= ****** GX
P.END = ****** GX      SERSLN= ****** GX       S.DEV   000020          $DNUMB= 000014          $RSXNM= ****** GX
P.FUDG= ****** GX      SERSLT  002162RG 002   S.DIA   000046          $EOS  = 000012          $SAVAL= ****** GX
P.NONE= ****** GX      SERST   000544R  003   S.DPA   000032          $ERRXA  000300RG 002    $STRNG= 000004
P.PAR1= ****** GX      SERTMP  002046RG 002   S.DPC   000036          $ERRXB  000324RG 002    $SUBXP= 000010
P.PAR2= ****** GX      SERUNI= ****** GX       S.DUM   000050          $ERRXC  000412RG 002    $TIOUT= ****** GX
P.UMR = ****** GX      SF.ACT= 000200          S.FLAG  000002          $ERRXD  000440RG 002    $$$FLG= 177777
QBUF    000644R        SF.CIR= 000010          S.FLG   000000          $ERRXE  000474RG 002    $$$KEY= 000033
QMFL    000660R        SF.DEV= 000020          S.HAD   000010          $ERRXF  000532RG 002    $$$STA= 000000
QPAR    000652R        SF.DIA= 004000          S.HST   000006          $ERRXG  000570RG 002    $$$TMP= 000267R  005
QROU  = ****** GX      SF.DPA= 000100          S.LEN   000004          $ERRXH  000614RG 002    .BRAD   002464RG  002
Q$$OPT= 000010         SF.DPC= 000200          S.LENG  000052          $ERRXI  000640RG 002    .FLBK0= ****** GX
RBRA  = 000076         SF.DUM= 010000          S.LNK   000000          $ERRXJ  000664RG 002    .FLBK1= ****** GX
READ    000552R        SF.ENA= 000100          S.LOA   000044          $ERRXK  000724RG 002    .FLBK2= ****** GX
REMDF   002006R  003   SF.HAD= 000004          S.NAME= ****** GX       $ERRXL  000754RG 002    .FLBK3= ****** GX
REPND   001174RG       SF.HST= 000002          S.NMST  000002          $ERRXM  001014RG 002    .FLBK4= ****** GX
REP.NN  001136RG       SF.LOA= 002000          S.OWNR  000003          $ERRXO  001114RG 002    .FUDGE= ****** GX
REP8B   001066RG       SF.LPB= 000004          S.PSS   000022          $ERRXP  001164RG 002    .NOMAP= ****** GX
REP8C   001066RG       SF.MFL= 000040          S.SEC   000040          $ERRXQ  001234RG 002    .RDBSZ= ****** GX
RETF  = ****** GX      SF.PAC= 000020          S.TER   000042          $ERRXR  001300RG 002    .TPARS= ****** GX
RTSPC   001044RG       SF.PH3= 020000

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001252    001   (RW,I,LCL,REL,CON)
DATA    002466    002   (RW,D,LCL,REL,CON)
$STATE  002116    003   (RW,D,LCL,REL,CON)
$KTAB   000070    004   (RW,D,LCL,REL,CON)
$KSTR   000277    005   (RW,D,LCL,REL,CON)
```

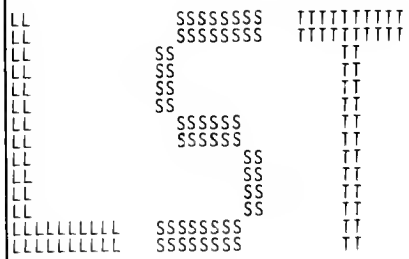

Symbol table

```
                .TITLE  NTLCFP - SCAN CONFIGURATION FILE FOR INITIALIZATION
                .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - CONFIG FILE SCAN FOR PSI PORTION OF "SET CEX"
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  15-JAN-81
;       VERSION 3.1 RELEASE
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
    495 000676                          20$:    RETURN
```

```
**FILE**ID**NTLCSR
```





LOCAL DATA

```
                                ;
000166                          FOUND:  .BLKW   1
                                ;
                                ; TKB BASE ADDRESS OF XXXTAB.TSK FILE
                                ;
000170                          BASE:   .BLKW   1
                                ;
                                ; NUMBER OF UNITS
                                ;
000172                          UNITS:  .BLKW   1
                                ;
                                ; END OF TABLES ADDRESS
                                ;
000174                          TABEND: .BLKW   1

                                .IFF    ; DF R$$11M

                                REQUES: RQST$   XX....          ; DPB TO REQUEST DEVICE DRIVER

                                DEVNAM: .ASCII  /XX./           ; FIRST THREE CHARACTERS OF DEVICE NAME

                                .EVEN
```

```
                        .SBTTL  $NLCDD - CALL DRIVER AT POWERFAIL ENTRY

                ;+
                ; $NLCDD - CALL DEVICE DRIVER AT POWERFAIL ENTRY (UNMAPPED ONLY)
                ;
                ; THIS ROUTINE IS USED TO CALL THE LLC'S DEVICE DRIVER AT HIS POWERFAIL
                ; ENTRY POINT FOR ALL UNITS WHICH HAVE THE UC.PWF BIT SET IN U.CTL.
                ;
                ; INPUTS:
                ;       R3=DCB ADDRESS
                ;       R4=NUMBER OF UNITS TO SCAN
                ;       R5=FIRST UCB ADDRESS
                ;
                ; OUTPUTS:
                ;       NONE
                ;-

                        .IF NDF M$$MGE

                $NLCDD:
                        SWSTK$  30$             ;* ENTER KERNEL MODE
                        MOV     R4,-(SP)        ;* SAVE UNIT COUNT
                10$:    BITB    #UC.PWF,U.CTL(R5) ;* CALL DRIVER AT POWERFAIL ENTRY?
                        BEQ     20$             ;* IF EQ, NO
                        MOV     R3,-(SP)        ;* SAVE DCB ADDRESS
                        MOV     R5,-(SP)        ;* AND UCB ADDRESS
                        MOV     D.DSP(R3),R2    ;* GET DISPATCH TABLE ADDRESS
                        MOV     U.SCB(R5),R4    ;* SCB ADDRESS
                        MOVB    S.CON(R4),R3    ;* AND CONTROLLER INDEX
                        CALL    @D.VPWF(R2)     ;* CALL DRIVER AT POWERFAIL ENTRY
                        MOV     (SP)+,R5        ;* RESTORE UCB ADDRESS
                        MOV     (SP)+,R3        ;* AND DCB ADDRESS
                20$:    MOV     $FEATR,U.CW3(R5) ;* SETUP THE FEATURES WORD
                        ADD     D.UCBL(R3),R5   ;* GIVING NEXT UCB ADDRESS
                        DEC     (SP)            ;* ALL DONE?
                        BNE     10$             ;* IF NE, NO
                        TST     (SP)+           ;* PURGE STACK
                30$:    RETURN                  ;* BACK TO USER MODE AND THEN TO CALLER

                        .ENDC
```

```
                                        .SBTTL  $MAPX - MAP TO EXTENDED MEMORY
                                ;+
                                ; $MAPX - Map to extended memory using one of the
                                ;         task's APRs (co-routine)
                                ;
                                ; Note - this routine must be called on kernel stack
                                ;
                                ; Inputs:
                                ;       R0=Extended memory bias
                                ;
                                ; Outputs:
                                ;       R1=Mapping register in use
                                ;       R2=Mapping bias to be used by caller
                                ;-
                                ;
000000                          $MAPX::
                                        .IF     NDF,S$$BAS&R$$PRO
                                
000000  016701  000000G                 MOV     ISAR5,R1        ;* KISAR5 is first task APR
000004  012702  120000                  MOV     #120000,R2      ;* Use APR5
                                
                                        .IFF
                                
                                        MOV     ISAR6,R1        ;* KISAR6 is second task APR
                                        MOV     #140000,R2      ;* Use APR6
                                
                                        .ENDC   ;NDF,S$$BAS&R$$PRO
                                
000010  010167  000000'                 MOV     R1,APR          ;* Set mapping register address
000014  011167  000002'                 MOV     (R1),APRSV      ;* Save mapping register contents
000020  010011                          MOV     R0,(R1)         ;* Map to extended memory
000022                                  CALL    @(SP)+          ;* Call the caller back
000024  016777  000002' 000000'         MOV     APRSV,@APR      ;* Restore mapping register contents
000032                                  RETURN                  ;* Back to user mode
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ALUN$	#5-64	5-64								
BDOFF$	#5-65									
CBYTE$	#5-64	5-64	5-64	#5-64	5-64					
CSI$	#5-64	5-64								
CWORD$	#5-64	5-64	5-64	5-64	#5-64	5-64	5-64	#5-64	5-64	5-64
DEF$L	#5-65									
DIR$	#5-64									
FDBDF$	#5-64	5-64								
FDBSZ$	#5-64	5-64								
FDOFF$	#5-64	5-64	#5-64	5-64						
FDOP$A	#5-64	5-64								
FDRC$A	#5-64	5-64								
FSRSZ$	#5-59	5-65								
GCLDF$	#5-59	5-64								
GCL$	#5-64									
GMCR$	#5-64	5-64								
NBOFF$	#5-64	5-64								
NMBLK$	#5-64	5-64								
OFF$	#5-64	#5-64	#5-64	#5-64	#5-64					
QDPB$	#5-64	5-64								
QIO$	#5-64	5-64								
RAD50$	#5-64	5-64	5-64							
SETF$	#5-64	5-64								
WTSE$	#5-64	5-64								

FILEID**NTLCEX

```
                                        .SBTTL  DECBLK - SET UP DECNET HOME BLOCK

                                ;+
                                ; DECBLK - SET UP DECNET HOME BLOCK POINTER
                                ;
                                ; INPUTS: NONE
                                ;
                                ; OUTPUT: $DECPT DECNET HOME BLOCK POINTER SET UP
                                ;
                                ;-

000552  010146                  DECBLK: MOV     R1,-(SP)        ; SAVE R1
000554  010246                          MOV     R2,-(SP)        ; SAVE R2
000556  012701  000104                  MOV     #D$END,R1       ; BLOCK SIZE TO ALLOCATE
000562  005000                          CLR     R0              ; ASSUME ALLOCATION FAILURE
000564                                  SWSTK$  20$             ; ENTER SYSTEM STATE
000570                                  CALL    @ALOCB          ; ALLOCATE DECNET HOME BLOCK
000574  103433                          BCS     10$             ; IF ERROR - BRANCH
000576  010066  000002                  MOV     R0,2(SP)        ; INDICATE SUCCESS
000602  105020                  5$:     CLRB    (R0)+           ; INIT BLOCK
000604  005301                          DEC     R1              ; ANY LEFT ?
000606  001375                          BNE     5$              ; IF YES - BRANCH
000610  016677  000002  000000G        MOV     2(SP),@DECPT    ; SET UP HOME BLOCK POINTER
000616  017700  000000G                 MOV     @DECPT,R0       ; ...
000622  010001                          MOV     R0,R1           ; SET UP REMOTE BLOCK LISTHEAD
000624  062701  000004                  ADD     #D$RNN+2,R1     ; ...
000630  010160  000002                  MOV     R1,D$RNN(R0)    ; ...
000634  112760  000252  000024          MOVB    #252,D$HIOR(R0) ; SET UP HI-ORDER BYTES OF ETHERNET ADDRESS
000642  112760  000000  000025          MOVB    #0,D$HIOR+1(R0) ; ...
000650  112760  000004  000026          MOVB    #4,D$HIOR+2(R0) ; ...
000656  112760  000000  000027          MOVB    #0,D$HIOR+3(R0) ; ...

000664                          10$:    RETURN
000666  012602                  20$:    MOV     (SP)+,R2        ; RESTORE R2
000670  012601                          MOV     (SP)+,R1        ; RESTROE R1
000672  005700                          TST     R0              ; HOME BLOCK ALLOCATION FAILURE ?
000674  001001                          BNE     25$             ; IF NOT - BRANCH
000676  000261                          SEC                     ; INDICATE ERROR
000700                          25$:    RETURN
```

```
 1050 002504  016767  000000G 000032'          MOV     .NBRA,NIRO      ; STORE ROUTER ADJACENCIES VALUE
 1051 002512  016767  000000G 000030'          MOV     .RPRI NIPR      ; STORE ROUTER PRIORITY VALUE
 1052
 1053 002520  016702  000034'                  MOV     XPTPDV,R2       ; GET XPT'S PDV ADDRESS
 1054
 1055 002524                                    SWSTK$  90$             ;; ENTER SYSTEM STATE
 1056
 1057 002530                                    SAVMAP                  ;; SAVE CURRENT MAPPING
 1058 002534  016202  000000G                  MOV     Z.DAT(R2),R2    ;; GET ADDRESS OF XPT'S DDB
 1059 002540  016203  000100                   MOV     N$TLC(R2),R3    ;; GET NUMBER OF COUNTERS
 1060 002544                                    MAP     N$TLC+2(R2)     ;; MAP TO COUNTER BLOCK
 1061 002552  016202  000104                   MOV     N$TLC+4(R2),R2  ;;GET VIRTUAL ADDRESS
 1062 002556                                    BIAS5   R2              ;; BIAS COUNTER BLOCK ADDRESS
 1063 002566  005712                  60$:     TST     (R2)            ;; HAS THIS BLOCK BEEN ASSIGNED ALREADY?
 1064 002570  100407                           BMI     70$             ;; BR IF NO - USE IT
 1065 002572  020412                           CMP     R4,(R2)         ;; ELSE IS THIS THE CORRECT ENTRY?
 1066 002574  001406                           BEQ     80$             ;; BR IF YES
 1067 002576  062702  000046                   ADD     #T$LLEN,R2      ;; POINT TO NEXT COUNTER BLOCK
 1068 002602                                    SOB     R3,60$          ;; CHECK ALL COUNTER BLOCKS
 1069 002606  000415                           BR      85$             ;; BR IF UNABLE TO ALLOCATE COUNTER BLOCK
 1070 002610  010412                  70$:     MOV     R4,(R2)         ;; STORE CHANNEL NUMBER AND PDV
 1071 002612  016762  000024' 000030  80$:     MOV     HTIM,T$T5(R2)   ;; STORE HELLO TIMER
 1072 002620  016762  000026' 000032           MOV     LTIM,T$T6(R2)   ;; STORE LISTEN TIMER
 1073 002626  016762  000032' 000042           MOV     NIRO,T$NRNI(R2);; STORE BROADCAST ROUTER ADJACENCIES
 1074 002634  016762  000030' 000040           MOV     NIPR,T$RPRI(R2);; STORE ROUTER PRIORITY
 1075
 1076 002642                          85$:     RESMAP                  ;; RESTORE PREVIOUS MAPPING
 1077 002646                                   RETURN
 1078
 1079 002650  062700  000002          90$:     ADD     #2,R0           ; UPDATE OFFSET FOR NEXT STATION ENTRY
 1080 002654  010304                           MOV     R3,R4           ; RESTORE STATION TABLE ADDRESS
 1081 002656  005301                           DEC     R1              ; MORE STATIONS TO CHECK?
 1082 002660  003270                           BGT     40$
 1083 002662  000241                  100$:    CLC                     ; SUCCESSFUL COMPLETION
 1084 002664                                   RETURN
 1085
 1086         000001                           .END
```

Table of contents

5-	58	MACRO DEFINITIONS
6-	79	CONSTANT DEFINITIONS
7-	91	LOCAL DATA
8-	164	ERROR MESSAGES
9-	210	$NLCFG - PROCESS CONFIGURATION FILE
10-	409	OPEN - OPEN THE CONFIGURATION FILE
11-	434	READ - READ NEXT RECORD
12-	464	Q*** - LOOK FOR ***$DF MACRO
13-	505	CERR - PRINT ERROR MESSAGE
13-	506	CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
14-	545	CFLIN - ECHO LAST RECORD
14-	546	FNLIN - ECHO FILE NAME AND LAST RECORD
15-	563	REP8A - REPLACE DEVICE NAME AND CONTROLLER NUMBER
16-	592	TPARS STATE TABLES - COMMON SUBEXPRESSIONS
17-	623	ADD$DF STATE TABLE
18-	633	LLC$DF STATE TABLE
19-	679	PLT$DF STATE TABLE
20-	690	COMMON ACTION ROUTINES
21-	738	ADD$DF ACTION ROUTINES
22-	751	PLT$DF ACTION ROUTINES
23-	768	GETCSR
24-	784	LLC$DF ACTION ROUTINES
25-	816	FNDPDV - FIND PDV
26-	851	LNKEND - LINK BLOCK AT END OF LIST

```
                                        .SBTTL  Q*** - LOOK FOR ***$DF MACRO

                                ;+
                                ; Q*** - LOOK FOR ***$DF MACRO
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3,R4,R5=DESTROYED
                                ;-

                                        .ENABL  LSB

001126  012705  000244'         QPLT:  MOV     #PLTDF,R5       ; STATE TABLE ADDRESS
001132  000405                         BR      10$

001134  012705  000100'         QADD:  MOV     #ADDDF,R5       ; STATE TABLE ADDRESS
001140  000402                         BR      10$

001142  012705  000114'         QLLC:  MOV     #LLCDF,R5       ; STATE TABLE ADDRESS

001146  005001                  10$:   CLR     R1              ; FULL KEYWORD MATCH LENGTH
001150  012702  000000'                MOV     #CFGKW,R2       ; KEYWORD TABLE ADDRESS
001154  016703  000352'                MOV     CFGSZ,R3        ; RECORD LENGTH
001160  012704  000144'                MOV     #CFGBF,R4       ; RECORD BUFFER ADDRESS
001164  005067  000356'                CLR     SYNERR          ; CLEAR SYNTAX ERROR FLAG
001170                                 CALL    .TPARS          ; GO DO THE PARSE
001174  103003                       BCC     20$             ; IF CC, FOUND WHAT WE WERE LOOKING FOR
001176  005367  000356'               DEC     SYNERR          ; DID SYNTAX ERROR OCCUR?
001202  001401                       BEQ     101$            ; IF EQ, YES
001204                          20$:   RETURN

                                ;
                                ; ERROR CONDITION
                                ;
001206                          101$:  EMSG$R  1T              ; SYNTAX ERROR

                                        .DSABL  LSB
```

```
                                        .SBTTL  FNDPDV - FIND PDV

                                ;+
                                ; FNDPDV - TRY TO FIND PDV
                                ;
                                ; INPUTS:
                                ;       R0=RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=PDV ADDRESS
                                ;       R1=PDV INDEX
                                ;       R2=DESTROYED
                                ;-
001744                          FNDPDV::
001744  010046                         MOV     R0,-(SP)        ;; SAVE NAME TO BE MATCHED
001746  017701  000000G                MOV     @PDVTA,R1       ;; GET PDV TABLE ADDRESS
                                ;
001752  017702  000000G                MOV     @PDVNM,R2       ;; AND NUMBER OF PDV'S
001756  001407                         BEQ     30$             ;; IF ZERO, NO PDV'S!
001760  012100                  10$:    MOV     (R1)+,R0        ;; GET PDV ADDRESS
001762  001403                         BEQ     20$             ;; IF ZERO, SKIP THIS ONE
                                ;
001764  026016  000000G                CMP     Z.NAM(R0),(SP)  ;; DOES THE NAME MATCH?
001770  001403                         BEQ     40$             ;; IF EQ, YES (C-BIT CLEAR)
001772                          20$:    SOB     R2,10$          ;; LOOP $PDVNM TIMES
001776  000261                  30$:    SEC                     ;; INDICATE FAILURE
002000  005226                  40$:    INC     (SP)+           ;; PURGE STACK
002002  006016                         ROR     (SP)            ;; SAVE C-BIT
                                ;
002004  005741                         TST     -(R1)           ;; CALCULATE PDV INDEX
002006  167701  000000G                SUB     @PDVTA,R1       ;;
002012  006316                         ASL     (SP)            ;; RESTORE C-BIT
002014                                  RETURN
```

```
                .TITLE  NTLCF2 - SCAN CONFIGURATION FILE FOR INITIALIZATION
                .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - CONFIG FILE SCAN FOR "SET CEX"
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```


MACRO DEFINITIONS

```
                                        .SBTTL  CFLIN - ECHO THE LAST RECORD READ
                                        .SBTTL  FNLIN - ECHO FILE NAME AND LAST RECORD READ

                                ;+
                                ;
                                ; CFLIN - ECHO THE LAST RECORD READ FROM THE CONFIG FILE
                                ; FNLIN - ECHO FILE NAME AND LAST RECORD READ FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
001046                          FNLIN:: CALL    $FNOUT          ; PRINT FILE NAME
001052  012700  002236'         CFLIN:: MOV     #CFGBF,R0       ; GET RECORD BUFFER ADDRESS
001056  016701  002444'                 MOV     CFGSZ,R1        ; AND RECORD SIZE
001062                                  CALLR   $TIOUT          ; PRINT IT ON TI:
```

```
                                        .SBTTL  TPARS STATE TABLES - LLC$DF,DLC$DF,DDM$DF

;
; LLC$DF, DLC$DF, DDM$DF
;
001252          STATE$  PRODF
001252           TRAN$   $RAD50,,PRONM
001252          STATE$
001252           TRAN$   <',>
001252          STATE$  PRODF1
001252           TRAN$   '!,PRODF1
001252           TRAN$   <',>,PRODF2
001252           TRAN$   !END,$EXIT
001252           TRAN$   %ZF.MFL%,PRODF1,PRMFL
001252           TRAN$   $RAD50,PRODF1
001252          STATE$  PRODF2
001252           TRAN$   $NUMBR
001252          STATE$
001252           TRAN$   !END,$EXIT
001252           TRAN$   <',>
001252          STATE$
001252           TRAN$   $NUMBR
001252          STATE$
001252           TRAN$   !END,$EXIT
001252           TRAN$   <',>
001252          STATE$
001252           TRAN$   $LAMDA,$EXIT
```

```
NTLCF2 - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:24  Page 32-3
Symbol table

Errors detected:  0

*** Assembler statistics


Work  file  reads: 36
Work  file writes: 34
Size of work file: 17918 Words  ( 70 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:04:36.58
SY:NTLCF2.V2,[132,134]NTLCF2/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLCF2
```



```
NTLCF2     CREATED BY  MACRO  ON 29-JUN-85 AT 00:27     PAGE 1
```

```
                                .SBTTL  MACRO DEFINITIONS

                        ;
                        ; LIBRARY MACROS
                        ;
                                .MCALL  ASL$,EMSG$R,GCL$,NTLDF$,NTLER$
                                .MCALL  SLTDF$,RETC

                        ;
                        ; LIBRARY SYMBOLS
                        ;
000000                          NTLDF$                  ; NETLDR REQUEST BLOCK OFFSETS
000000                          SLTDF$                  ; DEFINE SLT OFFSETS
```

```
                                        .SBTTL  PCKBCD - PACK STRING OF DIGITS IN BCD FORMAT
                                ;+
                                ;
                                ; PCKBCD - PACK A STRING OF DIGITS IN BCD FORMAT
                                ;
                                ; INPUTS:
                                ;       R0 - NUMBER OF BYTES IN PACKED ADDRESS
                                ;       R4 - ADDRESS OF BUFFER CONTAINING ASCI DIGITS TO PACK (0=END OF BUFFER)
                                ;       R5 - ADDRESS OF BUFFER TO STORE PACKED DATA
                                ;       COUNT - NUMBER OF DIGITS SPECIFIED
                                ;
                                ; OUTPUTS:
                                ;       BUFFER SPECIFIED IN R5 CONTAINS DATA IN BCD FORMAT
                                ;
                                ;-
                                PCKBCD::MOV     R5,R1                   ; GET ADDRESS OF PACKED DATA
000702  116702  000356'                 MOVB    COUNT,R2                ; GET NUMBER OF DIGITS IN STRING
000706  001423                          BEQ     40$                     ; BR IF NOTHING TO DO
000710  105021                  5$:     CLRB    (R1)+                   ; INITIALIZE PACKED ADDRESS
000712  005300                          DEC     R0                      ; MORE TO INITIALIZE?
000714  003375                          BGT     5$                      ; BR IF YES
000716  005001                          CLR     R1                      ; POSITION INIDCATOR (0=HIGH FOUR BITS)
000720  005305                          DEC     R5                      ; UPDATE ADDRESS FOR FIRST TIME
000722  112400                  10$:    MOVB    (R4)+,R0                ; GET DIGIT TO PACK
000724  005701                          TST     R1                      ; STORE IN HIGH FOUR BITS?
000726  001007                          BNE     20$                     ; BR IF NO
000730                                  ASL$    4,R0                    ; MOVE TO HIGH FOUR BITS
000740  005201                          INC     R1                      ; INDICATE STORE IN LOW FOUR BITS NEXT
000742  005205                          INC     R5                      ; POINT TO NEXT BYTE
000744  000401                          BR      30$                     ; BR TO STORE DECIMAL
000746  005301                  20$:    DEC     R1                      ; INDICATE STORE IN HIGH FOUR BITS NEXT
000750  150015                  30$:    BISB    R0,(R5)                 ; PACK THE NUMBER INTO THE BUFFER
000752  005302                          DEC     R2                      ; MORE TO CONVERT?
000754  003362                          BGT     10$                     ; BR IF YES
000756                          40$:    RETURN
```

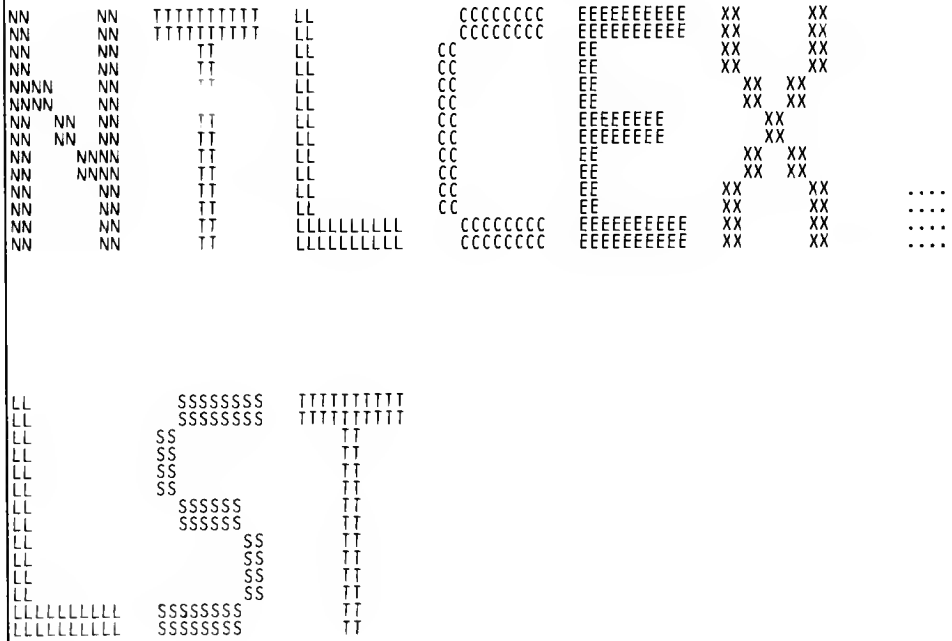
```
                        .TITLE  NTLCSR - CHECK CSR ADDRESS
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND   SHOULD   NOT   BE   CONSTRUED   AS   A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - CSR OFFLINE CHECKING ROUTINE
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

```
                                        .SBTTL  ERROR MESSAGES
                                ;
                                ; ERROR MESSAGES
                                ;
                                        .IFTF
                                        .ENABL  LC
                                        .IFT    ; DF R$$11M
000176                                  NTLER$  ,2A,4A,DVERR,REP4A,,<Undefined>
000220                                  NTLER$  ,2B,4,DVERR,$PRV.N,$FNOUT,<Open Failure (-***.)>
000256                                  NTLER$  ,2C,4,DVERR,$PRV.N,$FNOUT,<Read Failure (-***.)>
000314                                  NTLER$  ,2D,4,DVERR,$REP.P,$FNOUT,<Read Overflow>
000342                                  NTLER$  ,2E,4,DVERR,$REP.P,$FNOUT,<Format Error>
000370                                  NTLER$  ,2F,6,DVERR,$PRV.N,$FNOUT,<Open Failure (-***.)>
000426                                  NTLER$  ,2H,6,DVERR,$REP.C,$FNDEQ,<Not Contiguous>
000456                                  NTLER$  ,2J,6,DVERR,$CUR.N,$FNDEQ,<Label Block Read Failure (-***.)>
000530                                  NTLER$  ,2K,6,DVERR,$REP.C,$FNDEQ,<Not a Valid Binary Image>
000572                                  NTLER$  ,2L,9,DVERR,REP9,$CLDEQ,<Allocation Failure>
000626                                  NTLER$  ,2M,6,DVERR,$CUR.N,FNDEA,<Read Failure (-***.)>
000664                                  NTLER$  ,2N,4A,DVERR,REP4A,,<Odd>
000700                                  NTLER$  ,2P,4A,DVERR,REP4A,,<Outside Table Limits>
000736                                  NTLER$  ,2Q,9,DVERR,REP9,DVDEA,<Illegal Unit Count>
000772                                  NTLER$  ,2R,9,DVERR,REP9,DVDEA,<D.UCB Field Outside Table Limits>
001044                                  NTLER$  ,2S,9,DVERR,REP9,DVDEA,<UCB Address Odd>
001074                                  NTLER$  ,2T,9,DVERR,REP9,DVDEA,<UCB Address Outside Table Limits>
001146                                  NTLER$  ,2U,9,DVERR,REP9,DVDEA,<U.SCB Field Outside Table Limits>
001220                                  NTLER$  ,2V,9,DVERR,REP9,DVDEA,<SCB Address Odd>
001250                                  NTLER$  ,2W,9,DVERR,REP9,DVDEA,<SCB Address Outside Table Limits>
001322                                  NTLER$  ,2X,9,DVERR,REP9,,<Driver Already Resident>
001362                                  NTLER$  ,2Y,4A,DVERR,REP4A1,,<Undefined>

                                        .IFF    ; DF R$$11M

                                        NTLER$  ,2A,,DVERR,REPDV,,<Unable to run task **....>
                                        NTLER$  ,2B,,DVERR,REPDV,,<Device **: not in system>

                                        .ENABL  LC
                                        .NLIST  BEX
                                FMTO:   .ASCIZ  ""
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX
                                        .IFTF   ; DF R$$11M

                                        .DSABL  LC

                                        .IFT    ; DF R$$11M
                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
001404  052     040     123     FMT4:   .ASCIZ  "* Symbol Table File -- "
001434  052     040     123     FMT4A:  .ASCIZ  "* Symbol Table File -- Symbol * "
```

```
                                        .SBTTL  DVERR - PRINT ERROR MESSAGE

                                ;+
                                ; DVERR - PRINT ERROR MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;-
                                        .IFTF   ; DF    R$$11M

001324                          DVERR:  CALL    $SAVAL          ; SAVE ALL REGISTERS
001330  010005                          MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
001332  012704  000017'                  MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
001336  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
001340                                  CALL    @(R5)+          ; DO THE FORMATTING
001342  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
001344  001403                          BEQ     20$             ; IF EQ, NO
001346  112324                  10$:    MOVB    (R3)+,(R4)+     ; COPY THE REST OF THE FORMAT STRING
001350  001376                          BNE     10$             ;       ...
001352  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001354  012503                  20$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
001356  112524                  30$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
001360  001376                          BNE     30$             ;       ...
001362  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001364  012700  000007'                  MOV     #ERRMSG,R0      ; PRINT THE ERROR MESSAGE
001370  010401                          MOV     R4,R1           ;       ...
001372  160001                          SUB     R0,R1           ;       ...
001374                                  CALL    $TIOUT          ;       ...
001400  005703                          TST     R3              ; IS THERE A LINE 2 ADDRESS?
001402  001401                          BEQ     40$             ; IF EQ, NO
001404                                  CALL    (R3)            ; GO PRINT SECOND LINE
001406  000261                  40$:    SEC                     ; RETURN C-BIT SET
001410                                  RETURN
```

```
                                        .SBTTL  $MAPXP - MAP TO EXTENDED POOL
                                ;+
                                ;
                                ;       **-$MAPXP-Map to extended pool
                                ;
                                ;       Inputs:
                                ;               R2 -
                                ;
                                ;       Outputs:
                                ;               R2 - virtual address
                                ;               KISAR5 or (KISAR6-server base) mapped to extended pool
                                ;-
                                ;
000034  032702  000001  $MAPXP::BIT     #1,R2           ; Is the block in extended pool ?
000040  001424                 BEQ     100$            ; If EQ, no - already mapped to it
000042                          SAVRG   <R0>            ; Else, save R0
000044  010200                 MOV     R2,R0           ; Copy block address
000046  042702  177701         BIC     #177701,R2      ; Return virtual address
                                .IF     NDF,S$$BAS&R$$PRO
000052  052702  120000         BIS     #120000,R2      ; ... Mapped through KISAR5
                                .IFF    ;NDF,S$$BAS&R$$PRO
                                BIS     #140000,R2      ; ...Mapped through KISAR6
                                .ENDC   ;NDF,S$$BAS&R$$PRO
                                .REPT   6
                        000006  ROR     R0              ; Convert addres' to block bias
                                .ENDR
000072  042700  176000          BIC     #176000,R0      ; Clear extran. us bits
                                .IF     NDF,S$$BAS&R$$PRO
000076  017777  000000G 000000G MOV     @XBIAS,@ISAR5   ; Map to base address of pool
000104  060077  000000G         ADD     R0,@ISAR5       ; ... Update to proper block
                                .IFF    ;NDF,S$$BAS&R$$PRO
                                MOV     @XBIAS,@ISAR6   ; Map to base address of pool
                                ADD     R0,@ISAR6       ; ... Update to proper block
                                .ENDC   ;NDF,S$$BAS&R$$PRO
0001 0                          RESRG   <R0>            ; Restore R0
000112                  100$:   RETURN                  ; Return to the caller
                        000001  .END
```

```
**FILE**ID**NTLCEX
```


NN NN TTTTTTTTTT LL CCCCCCCC EEEEEEEEEE XX XX
NN NN TTTTTTTTTT LL CCCCCCCC EEEEEEEEEE XX XX
NN NN TT LL CC EE XX XX
NN NN TT LL CC EE XX XX
NNNN NN TT LL CC EE XX XX
NNNN NN LL CC EE XX XX
NN NN NN TT LL CC EEEEEEEE XX
NN NN NN TT LL CC EEEEEEEE XX
NN NNNN TT LL CC EE XX XX
NN NNNN TT LL CC EE XX XX
NN NN TT LL CC EE XX XX ....
NN NN TT LL CC EE XX XX ....
NN NN TT LLLLLLLLLL CCCCCCCC EEEEEEEEEE XX XX ....
NN NN TT LLLLLLLLLL CCCCCCCC EEEEEEEEEE XX XX ....

NTLCEX - LOAD COMM EXEC IMAGE MACRO V05.03b Saturday 29-Jun-85 00:22
Table of contents

```
                                        .SBTTL  NLBLK - SET UP NETLDR DATA BLOCK
                                ;+
                                ; NLBLK - SETUP NETLDR DATA BLOCK
                                ;
                                ; INPUTS:
                                ;       R1=CETAB ALLOCATION ADDRESS
                                ;       R2=CETAB ALLOCATION LENGTH
                                ;       .HOME=NETLDR DATA BLOCK ADDRESS
                                ;       CEXPCB=COMM EXEC PCB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       DATA BLOCK INITIALIZED
                                ;-
000702                          NLBLK:  SWSTK$  20$             ;* ENTER KERNEL MODE
000706  016700  000000G                 MOV     .HOME,R0        ;* GET DATA BLOCK ADDRESS
000712  010103                          MOV     R1,R3           ;* COPY CETAB ALLOCATION ADDRESS
000714  010204                          MOV     R2,R4           ;* COPY CETAB ALLOCATION LENGTH
                                
000716                                  CALL    $MAPX           ;* MAP TO THE DATA BLOCK
000722  062702  000000G                 ADD     #.CXCSR,R2      ;* SKIP THE NTINIT STUFF
000726  005022                          CLR     (R2)+           ;*
                                
                                        .IF DF  R$$MPL
                                        CLR     (R2)+           ;* SKIP CEX PCB ADDRESS
                                        .IFF
000730  016722  000036'                 MOV     CEXPCB,(R2)+    ;* STORE CEX PCB ADDRESS
                                        .ENDC
                                
000734  012700  000120'                 MOV     #XBLK1,R0       ;* GET NONVECTORED VALIDATION BLOCK ADDRESSES
000740  016022  000010          10$:    MOV     10(R0),(R2)+    ;* STORE SYMBOL VALUE
000744  011000                          MOV     (R0),R0         ;* GET NEXT SYMBOL BLOCK ADDRESS
000746  001374                          BNE     10$             ;* IF NE, NOT AT END YET
                                
000750  012700  000156'                 MOV     #VXBLK1,R0      ;* GET VECTORED VALIDATION BLOCK ADDRESSES
000754  016022  000010          12$:    MOV     10(R0),(R2)+    ;* STORE SYMBOL VALUE
000760  011000                          MOV     (R0),R0         ;* GET NEXT SYMBOL BLOCK ADDRESS
000762  001374                          BNE     12$             ;* IF MORE - BRANCH
                                
000764  005022                  15$:    CLR     (R2)+           ;* ZERO THE REMAINDER OF BLOCK
000766  032702  000077                  BIT     #77,R2          ;*      ...
000772  001374                          BNE     15$             ;*      ...
000774  162702  000100                  SUB     #100,R2         ;* BACK TO START OF HOME BLOCK
                                
001000  010362  000000G                 MOV     R3,.CXALL(R2)   ;* STORE THE CETAB ALLOCATION ADDRESS
001004  010462  000000G                 MOV     R4,.CXLEN(R2)   ;* STORE THE CETAB ALLOCATION LENGTH
                                
001010                          20$:    RETURN
```

Symbol table

```
ALOCB = ****** GX       CF.SOM= 000010          C.FLG   000022          D$SER   000032          FS.LTM= 001000
A$$CHK= 000000          CF.SYN= 000040          C.FLG1  000022          D$SQRL  000052          FS.MNT= 004000
A$$CPS= 000000          CF.TIM= 000400          C.FLG2  000032          D$$BUG= 177514          FS.MSN= 014000
A$$PRI= 000000          CF.TRN= 000100          C.FNC   000010          D$$ISK= 000000          FS.REA= 001000
A$$TRP= 000000          CMODE = 140000          C.LIN   000006          D$$L11= 000001          FS.RET= 000000
BUFUMP= 172354          CM.CIR= 000002          C.LNK   000000          D$$YNC= 000000          FS.REZ= 003000
CBLK1   000040R  002    CM.FMT= 100000          C.MOD   000011          D$$YNM= 000000          FS.RLB= 002000
CBLK3   000054R  002    CM.HRD= 000002          C.NSP   000004          ELTIME= ****** GX       FS.RNG= 011000
CBLK4   000070R  002    CM.LIN= 000000          C.PRO   000042          ETAB1   000334R  002    FS.RST= 000000
CBLK5   000104R  002    CM.LOO= 000001          C.RSV   000002          ETAB2   000344R  002    FS.RTN= 001000
CB.CCB= 000002          CM.XLO= 000004          C.STA   000007          E$NBR   000014          FS.SET= 005000
CB.DDM= 000040          CP.DCF= 000040          C.STS   000012          E$NBS   000020          FS.SFC= 005000
CB.DLC= 000020          CP.HDL= 000007          C.URM   177776          E$NCR   000034          FS.SFR= 006000
CB.RDB= 000004          CP.PS = 177400          C.XACP  000004          E$NCS   000036          FS.SFS= 004000
CB.SDB= 000010          CP.PSI= 000200          C.XID   000035          E$NIC   000044          FS.SPW= 040000
CB.SLI= 000100          CP.XCF= 000100          C.XLEN  000044          E$NLEN  000050          FS.STM= 000000
CB.XLB= 000001          CP.2FR= 000030          C.XPLI  000040          E$NLLA  000012          FS.STP= 002000
CCBCT = ****** GX       CRC$OV= ****** GX       C.XPT   000034          E$NLNK  000000          FS.STR= 001000
CCBLH = ****** GX       CS.ABO= 000100          C.XSVC  000042          E$NML   000040          FS.TRM= 003000
CCBNM = ****** GX       CS.BRO= 000002          C.XTC   000037          E$NMR   000024          FS.WLB= 001000
CCBS    001736R         CS.BUF= 000200          C.X25   000036          E$NMS   000030          FS.XKL= 002000
CCBSZ = ****** GX       CS.CES= 000002          DECBLK  000552R         E$NNOD  000002          FS.XOF= 010000
CC.LLC= 000200          CS.CHN= 000010          DECPT = ****** GX       E$NRT   000042          FS.XON= 007000
CEAVL = ****** GX       CS.CMP= 000200          D$AMXC  000072          E$NRTP  000005          FS.ZER= 002000
CECLK   001354R         CS.DCR= 000400          D$AMXH  000074          E$NSEG  000010          F$$LVL= 000001
CENTI   001354R         CS.DEF= 000004          D$ANN   000000          E$NTIM  000046          GTBUF   002250R
CEPTR   001012R         CS.DEV= 000002          D$BRPR  000102          E$NUSE  000004          G$$TPP= 000000
CEPWR = ****** GX       CS.DIS= 000040          D$BRTM  000100          E$$STRT 000006          G$$TSS= 000000
CESYM   000324R         CS.ENA= 000001          D$DELF  000045          E$$XPR= 000000          G$$TTK= 000000
CETNAM  000012R  002    CS.ENB= 000020          D$DELW  000046          FC.CCP= 000020          G$$WRD= 000000
CEXNAM  000006R  002    CS.ERR= 100000          D$END = 000104          FC.CTL= 000006          HTIM    000024R  002
CEXPCB  000036R  002    CS.FTL= 001000          D$FNB   000034          FC.KCP= 000016          IBLK1   000214R  002
CE.ABO= 100362          CS.HCR= 000001          D$HIOR  000024          FC.KIL= 000004          IBLK2   000226R  002
CE.DAO= 100346          CS.HFE= 002000          D$HOST  000022          FC.MAN= 000024          IBLK3   000240R  002
CE.DIS= 100366          CS.LST= 040000          D$INAC  000044          FC.MLD= 000026          I$$RAR= 000000
CE.ERR= 100370          CS.MTL= 004000          D$INCT  000042          FC.PCT= 000030          I$$RDN= 000000
CE.ILN= 100350          CS.RNG= 000010          D$IPL   000051          FC.PWR= 000022          KISAR0= 172340
CE.LTO= 100356          CS.ROV= 000004          D$LID   000020          FC.RCE= 000002          KISAR6= 172354
CE.MOP= 100372          CS.RSN= 010000          D$LNAM  000006          FC.RCP= 000014          KSAR5 = ****** GX
CE.NTE= 100361          CS.SHU= 000001          D$LNUM  000014          FC.TIM= 000010          K$$CNT= 177546
CE.RTE= 100376          CS.SID= 000002          D$LST   000047          FC.XCP= 000012          K$$CSR= 177546
CE.SRC= 100364          CS.STR= 000004          D$MAXC  000064          FC.XME= 000000          K$$LDC= 000000
CE.STP= 100352          CS.SUC= 000001          D$MAXH  000066          FE.CEX= ****** GX       K$$TPS= 000074
CE.TME= 100354          CS.TMO= 020000          D$MAXV  000070          FL.NTI= ****** GX       K6TMP   000000R  002
CE.TMO= 100374          CS.XUR= 000004          D$MLL   000040          FMASK = ****** GX       LD$LP = 000000
CE.UNS= 100344          CXOPT = ****** GX       D$MNOD  000041          FS.AST= 000000          LF.ACT= 100000
CF.CHN= 000001          C$$CKP= 000000          D$NA    000062          FS.CIB= 002000          LF.BRO= 000400
CF.DDM= 000002          C$$ORE= 000400          D$NBEA  000056          FS.CRA= 001000          LF.BWT= 000007
CF.DYN= 000004          C$$RSH= 177564          D$NBRA  000054          FS.DIS= 013000          LF.ENA= 002000
CF.EIS= 000010          C.ADD   000034          D$NEND= 000054          FS.DVC= 001000          LF.LPB= 001000
CF.EOM= 000004          C.BID   000003          D$NLN   000030          FS.ENB= 012000          LF.MDC= 000100
CF.FRK= 100000          C.BUF   000014          D$NN    000060          FS.EXI= 001000          LF.MFL= 004000
CF.HDR= 000020          C.BUF1  000014          D$OUTT  000043          FS.GET= 006000          LF.MTP= 000020
CF.LB = 100000          C.BUF2  000024          D$RETF  000050          FS.HLT= 000000          LF.PAC= 000200
CF.LIN= 000002          C.CNT   000020          D$RNN   000002          FS.INI= 000000          LF.RDY= 040000
CF.LOG= 000020          C.CNT1  000020          D$RTMR  000076          FS.KIL= 000000          LF.REA= 010000
CF.MDM= 000001          C.CNT2  000030          D$SEG   000036          FS.LCL= 100000          LF.SER= 000040
```


Symbol table

```
                        .TITLE  NTLCFG - SCAN CONFIGURATION FILE FOR DYNAMIC INFORMATION
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - CONFIG FILE SCAN FOR "SET LINE" AND "SET PROCESS"
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.01  27-Sep-82
;       Add support for PLT$DF macro. (number of EPM channels)
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                        .SBTTL  CERR - PRINT ERROR MESSAGE
                                        .SBTTL  CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
                                ;+
                                ; CERR - PRINT ERROR MESSAGE
                                ; CFERR - PRINT FILE ASSOCIATED ERROR MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;       RETURN TO $NLCFG'S CALLER
                                ;-
001214  010046                  CERR::  MOV     R0,-(SP)        ; SAVE R0
001216                                  CALL    $CLDEQ          ; CLOSE THE CONFIG FILE
001222  012600                          MOV     (SP)+,R0        ; RESTORE R0
001224  010005                  CFERR:: MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
001226  012704  000027'                 MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
001232  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
001234                                  CALL    @(R5)+          ; DO THE FORMATTING
001236  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
001240  001403                          BEQ     20$             ; IF EQ, YES
001242  112324                  10$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
001244  001376                          BNE     10$             ;       ...
001246  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001250  012503                  20$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
001252  112524                  30$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
001254  001376                          BNE     30$             ;       ...
001256  005304                          DEC     R4              ; SKIP THE ZERO BYTE
001260  012700  000017'                 MOV     #ERRMSG,R0      ; GET ERROR MESSAGE ADDRESS
001264  010401                          MOV     R4,R1           ; COPY END ADDRESS
001266  160001                          SUB     R0,R1           ; GIVING LENGTH
001270                                  CALL    $TIOUT          ; PRINT IT
001274  005703                          TST     R3              ; SHOULD SECOND LINE BE PRINTED?
001276  001401                          BEQ     40$             ; IF EQ, NO
001300                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
001302  016706  000354'         40$:    MOV     SPSAV,SP        ; RESTORE SP FROM ENTRY
001306  000261                          SEC                     ; RETURN C-BIT SET
001310                          RTSPC:: RETURN
```

```
                                        .SBTTL  LNKEND - LINK BLOCK AT END OF LIST

                                ;+
                                ;
                                ; LNKEND - LINK BLOCK AT END OF LIST
                                ;
                                ; INPUTS:
                                ;       R0 - ADDRESS OF LISTHEAD
                                ;       R5 - UNMAPPED ADDRESS OF CURRENT BLOCK
                                ;
                                ; OUTPUTS:
                                ;       BLOCK IS LINKED AT END OF LIST
                                ;
                                ;-
002016  010001                  LNKEND::MOV     R0,R1                   ;; SAVE ADDRESS OF PREVIOUS BLOCK
002020  011000                          MOV     (R0),R0                 ;; GET ADDRESS OF NEXT BLOCK IN LIST
002022  001405                          BEQ     10$                     ;; BR IF END OF LIST - LINK IT HERE
002024  010046                          MOV     R0,-(SP)                ;; SET UP UNMAPPED ADDR FOR CONVERSION
002026                                  CALL    $CEACX                  ;; CONVERT TO MAPPED ADDRESS
002032  012600                          MOV     (SP)+,R0                ;; RETRIEVE MAPPED ADDRESS
002034  000770                          BR      LNKEND                  ;; GET NEXT BLOCK IN LIST
002036  010546                  10$:    MOV     R5,-(SP)                ;; GET UNMAPPED ADDR OF CURRENT BLOCK
002040  010146                          MOV     R1,-(SP)                ;; GET ADDRESS OF PREVIOUS BLOCK
002042                                  CALL    $LINKX                  ;; LINK BLOCK INTO LIST
002046                                  RETURN
```

```
                                        .SBTTL  MACRO DEFINTIONS
                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  ASL$,EMSG$R,GCL$,ISTAT$,NTLDF$,NTLER$,STATE$,TRAN$
                                        .MCALL  SLTDF$,DHBDF$,FNBDF$,SERDF$

                                        .IF DF, R$$11M
                                        .MCALL  EMSG$
                                        .ENDC

                                ;
                                ; LIBRARY SYMBOLS
                                ;
000000                                  NTLDF$                  ; NETLDR REQUEST BLOCK OFFSETS
000000                                  DHBDF$                  ; DECNET HOME BLOCK OFFSET
000000                                  SLTDF$                  ; DEFINE SLT OFFSETS
000000                                  FNBDF$                  ; DEFINE DLL FILE BLOCK OFFSETS
000000                                  SERDF$                  ; DEFINE SERVER BLOCK OFFSETS
```

```
                                        .SBTTL  REP8B - REPLACE PARTITION NAME
                                        .SBTTL  REP8C - REPLACE REGION NAME

                                ;+
                                ;
                                ; REP8B - REPLACE PARTITION NAME
                                ; REP8C - REPLACE REGION NAME
                                ;
                                ; INPUTS:
                                ;       R2 = POINTER TO RAD50 NAME
                                ;       R3 = FORMAT STRING ADDRESS
                                ;       R4 = ERROR MESSAGE BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2 = DESTROYED
                                ;       R3,R4 = UPDATED
                                ;
                                ;-

001066                          REP8B::
001066  112314                  REP8C:: MOVB    (R3)+,(R4)      ; COPY FORMAT STRING
001070  122724  000052                  CMPB    #'*,(R4)+       ; LOOKING FOR THE ASTERISK
001074  001374                          BNE     REP8B           ;
001076  005304                          DEC     R4              ; SKIP THE ASTERISK
001100  010400                          MOV     R4,R0           ; COPY BUFFER ADDRESS
001102  012201                          MOV     (R2)+,R1        ; GET FIRST HALF OF NAME
001104  010246                          MOV     R2,-(SP)        ; SAVE POINTER TO NAME
001106                                  CALL    $C5TA           ; CONVERT TO ASCII
001112  013601                          MOV     @(SP)+,R1       ; GET SECOND HALF OF NAME
001114  001402                          BEQ     10$             ; IF ZERO, IT'S ALL SPACES
001116                                  CALL    $C5TA           ; CONVERT SECOND HALF
001122  122740  000040          10$:    CMPB    #' ,-(R0)       ; SKIP TRAILING BLANKS
001126  001775                          BEQ     10$             ;
001130  005200                          INC     R0              ;
001132  010004                          MOV     R0,R4           ; SET NEW BUFFER ADDRESS
001134                                  RETURN
```

```
                                        .SBTTL  TPARS STATE TABLES - SLT$DF
                                ;
                                ; SLT$DF
                                ;
001252                          STATE$  SLTDF
001252                           TRAN$   $RAD50,,S.NAME
001252                          STATE$
001252                           TRAN$   <',>
001252                          STATE$
001252                           TRAN$   $RAD50
001252                          STATE$
001252                           TRAN$   <',>
001252                          STATE$
001252                           TRAN$   $RAD50
001252                          STATE$
001252                           TRAN$   <',>
001252                          STATE$
001252                           TRAN$   <',>,SLTDF2
001252                           TRAN$   $LAMDA
001252                          STATE$  SLTDF3
001252                           TRAN$   %LF.MFL%,SLTDF1,,LF.MFL,MRKFL
001252                           TRAN$   %LF.ENA%,SLTDF1,,LF.ENA,MRKFL
001252                           TRAN$   %LF.SER%,SLTDF1,,LF.SER,MRKFL
001252                           TRAN$   $RAD50
001252                          STATE$  SLTDF1
001252                           TRAN$   '!,SLTDF3
001252                           TRAN$   <',>
001252                          STATE$  SLTDF2
001252                           TRAN$   $NUMBR,,S.CTL
001252                          STATE$
001252                           TRAN$   <',>
001252                          STATE$
001252                           TRAN$   $NUMBR,,S.UNT
001252                          STATE$
001252                           TRAN$   !END,$EXIT
001252                           TRAN$   <',>
001252                          STATE$
001252                           TRAN$   <',>,SCTIM
001252                           TRAN$   $STRNG
001252                          STATE$
001252                           TRAN$   !END,$EXIT
001252                           TRAN$   <',>
001252                          STATE$  SCTIM
001252                           TRAN$   $NUMBR,SCHELO,S.CTIM
001252                           TRAN$   <',>,SCHEL
001252                          STATE$  SCHELO
001252                           TRAN$   !END,$EXIT
001252                           TRAN$   <',>
001252                          STATE$  SCHEL
001252                           TRAN$   $NUMBR,BRADJ
001252                           TRAN$   <',>
001252                          STATE$  BRADJ
001252                           TRAN$   <',>,BRADJ1
001252                           TRAN$   !END,$EXIT
001252                          STATE$  BRADJ1
001252                           TRAN$   $NUMBR,$EXIT,S.BRA
```

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
BUFDF	000574 R	11-493								
CERR	000742 RG	7-110	7-112	7-113	7-117	7-119	7-121	7-122	7-129	7-132
		7-133	7-134	7-135	7-141	7-142	7-143	7-144	7-145	7-146
		7-147	7-148	7-149	7-150	7-151	#12-541			
CFERR	000752 RG	7-107	7-108	7-109	7-127	7-128	#12-545			
CFGBF	002236 RG	#7-229	10-459	11-503	13-590					
CFGEXT	000006 R	#6-89	9-433							
CFGKW	000000 RG	11-501	#17-701							
CFGNAM	000000 R	#6-88	9-432							
CFGST	000000 RG	#17-701								
CFGSZ	002444 RG	#7-230	*10-461	11-502	13-591					
CFLIN	001052 RG	7-110	7-112	7-113	7-117	7-119	7-121	7-122	7-129	7-132
		7-133	7-134	7-135	7-142	7-143	7-144	7-145	7-146	7-147
		7-148	7-149	7-150	7-151	#13-590				
CIRMUL	002056 RG	#7-197								
CIRNAM	002050 RG	#7-195								
CIRNUM	002054 RG	#7-196								
CIRUNT	002154 RG	#7-211								
CWARN	000732 RG	7-114	#12-537							
DEACB	= ****** GX	8-349								
DECPT	= ****** GX	8-334	8-346	8-348	16-680					
DEVBUF	002142 RG	#7-208	7-209							
DEVNM	002060 RG	#7-198								
DVSZ	= 000010 G	#7-209								
D$END	= 000104	8-347								
D$LNAM	000006	8-339	16-681							
D$NBRA	000054	*8-335	*8-338							
D$NN	000060	8-336								
ECLLLN	002460 RG	#7-256								
ECLNOD	002462 RG	#7-257								
ENDHEX	002104 RG	#7-200								
ENDPAS	002032 RG	#7-189								
ERRBUF	000022 R	#6-96	12-548							
ERRMSG	000012 R	#6-94	12-560							
FIELDC	= 000003 G	#7-223								
FILBUF	002122 RG	#7-205	7-206							
FILFLG	002152 RG	#7-210								
FLSZ	= 000020 G	#7-206								
FL.NTI	= ****** GX	8-290								
FMT8	001726 RG	7-107	7-108	7-109	7-110	7-122	7-127	7-128	7-129	7-132
		7-134	7-135	7-141	7-142	7-143	7-144	7-145	7-146	7-147
		7-148	7-149	7-150	7-151	#7-161				
FMT8B	001746 RG	7-112	7-113	7-117	7-119	7-121	7-133	#7-162		
FMT9	001710 RG	7-114	#7-160							
FM.8	= 000000	#7-107	#7-108	#7-109	#7-110	#7-122	#7-127	#7-128	#7-129	#7-132
		#7-134	#7-135	#7-141	#7-142	#7-143	#7-144	#7-145	#7-146	#7-147
		#7-148	#7-149	#7-150	#7-151					
FM.8B	= 000000	#7-112	#7-113	#7-117	#7-119	#7-121	#7-133			
FM.9	= 000000	#7-114								
FNBEXT	= 000006 G	#7-222								
FNBLN	002164 RG	#7-216								
FNBSZ	= 000044 G	#7-219								

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 2

```
                                        .SBTTL  LOCAL SYMBOL DEFINITIONS

                                ;
                                ; LOCAL SYMBOL DEFINITIONS
                                ;
000074                                  LBRA = '<               ; LEFT ANGLE BRACKET
000076                                  RBRA = '>               ; RIGHT ANGLE BRACKET
000040                                  SPACE = 40              ; ASCII SPACE
000040                                  BLKSMN = 32.            ; MINIMUM VALUE FOR BLOCK SIZE
002000                                  BLKSMX = 1024.          ; MAXIMUM VALUE FOR BLOCK SIZE
007777                                  CHNLMX == 4095.         ; MAXIMUM VALUE FOR CHANNEL NUMBER
000006                                  CUNMMX == 6             ; MAXIMUM VALUE FOR CUG NAME
000000                                  SUBAMN == 0             ; MINIMUM VALUE FOR SUBADDRESS RANGE
023417                                  SUBAMX == 9999.         ; MAXIMUM VALUE FOR SUBADDRESS RANGE
000177                                  WNDSMX = 127.           ; MAXIMUM VALUE FOR WINDOW SIZE
```

```
                                        .SBTTL  LNKEND - LINK BLOCK AT END OF LIST

                                ;+
                                ;
                                ; LNKEND - LINK BLOCK AT END OF LIST
                                ;
                                ; INPUTS:
                                ;       R0 - ADDRESS OF LISTHEAD
                                ;       R5 - UNMAPPED ADDRESS OF CURRENT BLOCK
                                ;
                                ; OUTPUTS:
                                ;       BLOCK IS LINKED AT END OF LIST
                                ;
                                ;-
000760  010001                  LNKEND::MOV     R0,R1                   ;; SAVE ADDRESS OF PREVIOUS BLOCK
000762  011000                          MOV     (R0),R0                 ;; GET ADDRESS OF NEXT BLOCK IN LIST
000764  001405                          BEQ     10$                     ;; BR IF END OF LIST - LINK IT HERE
000766  010046                          MOV     R0,-(SP)                ;; SET UP UNMAPPED ADDR FOR CONVERSION
000770                                  CALL    $CEACX                  ;; CONVERT TO MAPPED ADDRESS
000774  012600                          MOV     (SP)+,R0                ;; RETRIEVE MAPPED ADDRESS
000776  000770                          BR      LNKEND                  ;; GET NEXT BLOCK IN LIST
001000  010546                  10$:    MOV     R5,-(SP)                ;; GET UNMAPPED ADDR OF CURRENT BLOCK
001002  010146                          MOV     R1,-(SP)                ;; GET ADDRESS OF PREVIOUS BLOCK
001004                                  CALL    $LINKX                  ;; LINK BLOCK INTO LIST
001010                                  RETURN
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  DIR$,SVTK$
                                        .IF DF  R$$MPL
                                        .MCALL  STAF$,RMAF$S
                                        .ENDC

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                ;
                                ; SET SST VECTOR DPBS
                                ;
000000                          SST:    SVTK$   SSTAB,2
000006                          NOSST:  SVTK$

                                ; TEMPORARY STORAGE OF UISAR7,UISDR7
                                ;
000014                          UAR7SV: .BLKW   1
000016                          UDR7SV: .BLKW   1

                                ;
                                ; SST TRANSFER TABLE
                                ;
000020  000120' 000120'         SSTAB:  .WORD   OFFLIN,OFFLIN

                                ;
                                ; SET AFFINITY DPB
                                ;
                                        .IF DF  R$$MPL
                                SETAF:  STAF$   0
                                        .ENDC

000000                                  .PSECT
```

LOCAL DATA

```
    202 001475     052     040     111  FMT6:   .ASCIZ  "* Image File -- "
    203 001516     052     072     040  FMT9:   .ASCIZ  "*: Device Tables -- "
    204                                         .EVEN
    205                                         .DSABL  LC
    206                                         .LIST   BEX
    207
    208 000000                                  .PSECT  ..BUF,D
    209
    210                                 ;
    211                                 ; DISK BUFFER FOR LABEL BLOCK READ
    212                                 ;
    213 000000                          DSKBUF: .BLKW   1
    214
    215                                         .IFTF   ; DF R$$11M
    216
    217 000000                                  .PSECT
```

```
                                        .SBTTL  DVDEA - DEALLOCATE DATA STRUCTURES AND CLOSE FILE
                                        .SBTTL  FNDEA - PRINT FILE NAME AND DEALLOCATE
                                        .SBTTL  $FNDEQ - PRINT FILE NAME AND CLOSE FILE


                                ;+
                                ;
                                ; DVDEA - DEALLOCATE DATA STRUCTURES AND CLOSE IMAGE FILE
                                ; FNDEA - PRINT FILE NAME AND DE-ALLOCATE
                                ; $FNDEQ - PRINT FILE NAME AND CLOSE FILE
                                ;
                                ; INPUTS:
                                ;       $LBUF=BUFFER ADDRESS
                                ;       $LLEN=BUFFER LENGTH
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
                                        .IFT    ; DF    R$$11M
001412                          FNDEA:  CALL    $FNOUT          ; PRINT FILE NAME
001416                          DVDEA:  SWSTK$  $CLDEQ          ;* ENTER KERNEL MODE
001422  016700  000000G                 MOV     $LBUF,R0        ;* GET BUFFER ADDRESS
001426  016701  000000G                 MOV     $LLEN,R1        ;* AND LENGTH
001432                                  CALLR   @DEACB          ;* DE-ALLOCATE DEVICE DATA STRUCTURES
                                ;
001436                          $FNDEQ: CALL    $FNOUT          ; PRINT FILE NAME
001442                                  CALLR   $CLDEQ          ; CLOSE FILE AND EXIT
```

```
MAPX    MACRO V05.03b Sunday 30-Jun-85 00:00  Page 6-1
Symbol table

APR     000000RG    002 D$$YNM= 000000          K$$TPS= 000074          N$$ACC= 000001          R$$K11= 000001
APRSV   000002RG    002 E$$XPR= 000000          L0$LP = 000000          N$$BUF= 000001          R$$SND= 000000
A$$CHK= 000000          F$$LVL= 000001          L$$ASG= 000000          N$$LDV= 000001          R$$11M= 000000
A$$CPS= 000000          G$$TPP= 000000          L$$DRV= 000000          N$$MCP= 000001          S$$WRG= 000000
A$$PRI= 000000          G$$TSS= 000000          L$$P11= 000001          N$$MLL= 000001          S$$YSZ= 007600
A$$TRP= 000000          G$$TTK= 000000          L$$11R= 000000          N$$MOV= 000010          T$$KMG= 000000
C$$CKP= 000000          G$$WRD= 000000          M$$CRB= 000124          N$$NCT= 000001          T$$MIN= 000000
C$$ORE= 000400          ISAR5 = ****** GX       M$$CRX= 000000          N$$PEM= 000001          V$$CTR= 001000
C$$RSH= 177564          I$$RAR= 000000          M$$FCS= 000000          P$$P45= 000000          XBIAS = ****** GX
D$$BUG= 177514          I$$RDN= 000000          M$$MGE= 000000          P$$WRD= 000000          X$$DBT= 000000
D$$DSK= 000000          K$$CNT= 177546          M$$NET= 000000          Q$$OPT= 000010          $MAPX   000000RG    003
D$$L11= 000001          K$$CSR= 177546          M$$OVR= 000000          R$$DER= 000000          $MAPXP  000034RG    003
D$$SYNC= 000000         K$$LDC= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
DATA    000004    002   (RW,D,LCL,REL,CON)
.MAPX   000114    003   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9038 Words  ( 36 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.42
SY:MAPX.V2,[141,134]MAPX/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[141,10]MAPX
```



```
MAPX        CREATED BY  MACRO  ON 30-JUN-85 AT 00:00     PAGE 1
SYMBOL CROSS REFERENCE                                   CREF  04.00
```

Table of contents

5- 54 MACRO DEFINITIONS
6- 99 LOCAL DATA
7- 285 $SCEX - TRANSFER COMM EXEC INTO CORE
8- 381 CESYM - LOOK FOR SPECIAL SYMBOLS
9- 489 DECBLK - SET UP DECNET HOME BLOCK
10- 531 NLBLK - SET UP NETLDR DATA BLOCK
11- 582 CEPTR - RELOCATE CETAB ADDRESSES
12- 668 CECLK - SET UP MULTI-PROCESSOR CLOCK QUEUE ENTRIES
13- 702 CENTI - NTINIT PART OF SET SYS + NTINIT
14- 805 CCBS - ALLOCATE AND LINK CCB'S
15- 867 RDBS - ALLOCATE RDBS
16- 913 SDBS - ALLOCATE SDBS
17- 960 GTBUF - GET NEXT BUFFER ADDRESS
18- 993 $ALTLC - SET UP XPT CIRCUIT COUNTER BLOCK

```
                                        .SBTTL  CEPTR - RELOCATE CETAB ADDRESSES

                                ;+
                                ; CEPTR - RELOCATE CETAB ADDRESSES
                                ;
                                ; INPUTS:
                                ;       R1=CETAB ALLOCATION ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       $NTLPT=NETLDR DATA BLOCK ADDRESS
                                ;       R0,R1,R2=DESTROYED
                                ;
                                ;-
001012                          CEPTR:
001012  010146                          MOV     R1,-(SP)        ; SAVE R1
001014  017700  000000G                 MOV     @TKPS,R0        ; GET PROGRAMMABLE CLOCK VALUE
001020  012701  000012                  MOV     #10.,R1         ; CALCULATE $TKPS/10.
001024                                  CALL    $DIV            ; DO IT
001030  010077  000000G                 MOV     R0,@TK100       ; INITIALIZE TIMER VALUE
001034  012601                          MOV     (SP)+,R1        ; RESTORE R1
001036  016777  000262' 000000G         MOV     .PDVTB+10,@PDVTA ; INITIALIZE THE TABLE POINTERS
001044  016777  000274' 000000G         MOV     .SLTMB+10,@SLTMA ;      ...
001052  016777  000306' 000000G         MOV     .LLCTB+10,@LLCTA ;      ...
001060  016777  000320' 000000G         MOV     SLTTOT+10,@SLTNM ; INITIALIZE THE NUMBER OF TABLE ENTRIES
001066  016777  000332' 000000G         MOV     PDVTOT+10,@PDVNM ;      ...
001074  016777  000224' 000000G         MOV     IBLK1+10,@CXOPT ; SET UP COMM EXEC OPTIONS
001102  005077  000000G                 CLR     @LGDDB          ; NO LOGGING DATABASE YET
001106  017700  000000G                 MOV     @DECPT,R0       ; STORE DECNET HOME BLOCK
001112  016760  000000G 000060         MOV     .NN,D$NN(R0)    ; INITIALIZE ROUTING PARAMETERS
001120  016760  000000G 000076         MOV     .ROUTM,D$RTMR(R0) ;     ...
001126  016760  000000G 000064         MOV     .MAXC,D$MAXC(R0) ;      ...
001134  016760  000000G 000066         MOV     .MAXH,D$MAXH(R0) ;      ...
001142  016760  000000G 000070         MOV     .MAXH,D$MAXV(R0) ;      ...
001150  016760  000000G 000062         MOV     .NA,D$NA(R0)    ;       ...
001156  016760  000000G 000056         MOV     .NBEA,D$NBEA(R0) ;      ...
001164  016760  000000G 000072         MOV     .AMAXC,D$AMXC(R0) ;     ...
001172  016760  000000G 000074         MOV     .AMAXH,D$AMXH(R0) ;     ...
001200  016760  000000G 000100         MOV     .ROUTB,D$BRTM(R0) ;     ...
001206  006360  000070                 ASL     D$MAXV(R0)      ;       ...
001212  016777  000000G 000000G         MOV     .HOME,@NTLPT    ; COPY NETLDR DATA BLOCK ADDRESS

                                ;
                                ; RELOCATE CETAB ADDRESSES
                                ;

001220  060177  000000G                 ADD     R1,@PDVTA       ; RELOCATABLE ADDRESSES
001224  060177  000000G                 ADD     R1,@SLTMA       ;       ...
001230  060177  000000G                 ADD     R1,@LLCTA       ;       ...
001234  017700  000000G                 MOV     @PDVTA,R0       ; RELOCATE THE PDV TABLE
001240  017702  000000G                 MOV     @PDVNM,R2       ;       ...
001244  001440                          BEQ     101$            ; MUST BE AT LEAST 1 PDV !!!
001246  060120                  5$:     ADD     R1,(R0)+        ;       ...
001250                                  SOB     R2,5$           ; REPEAT $PDVNM TIMES
001254  017700  000000G                 MOV     @SLTMA,R0       ; RELOCATE THE SLT TABLE
001260  017702  000000G                 MOV     @SLTNM,R2       ;       ...
001264  001421                          BEQ     10$             ; IF EQ, NO ENTRIES IN SLT TABLE
```

Symbol table

```
LF.TIM= 000010          N$ADJ2  000074          RDBCT = ****** GX       T$LTCL  000024          $ERR56= ****** GX
LF.UNL= 020000          N$CACH  000062          RDBLH = ****** GX       T$LTIM  000026          $ERR57= ****** GX
LF.X2P= 000000          N$CRC   000120          RDBNM = ****** GX       T$LTPR  000014          $ERR58= ****** GX
LGDDB = ****** GX       N$HC1   000052          RDBS    002032R         T$LTPS  000020          $ERR59= ****** GX
LLCTA = ****** GX       N$HC2   000056          RDBSZ = ****** GX       T$NAPL  000004          $ERR60= ****** GX
LN.CLO= 000000          N$LV1   000002          RDBTH = ****** GX       T$NFE   000000          $ERR61= ****** GX
LN.DUM= 000005          N$LV2   000010          R$$DER= 000000          T$NLEN  000010          $ERR62= ****** GX
LN.LOA= 000004          N$MHC1  000036          R$$K11= 000001          T$NNUL  000002          $ERR63= ****** GX
LN.LOD= 000003          N$MHC2  000044          R$$SND= 000000          T$NOPL  000006          $ERR64= ****** GX
LN.OAU= 000003          N$PLD   000016          R$$T1M= 000000          T$NRNI  000042          $ERR65= ****** GX
LN.OFF= 000001          N$PRI   000076          SDBCT = ****** GX       T$NRPL  000005          $ERR66= ****** GX
LN.ON = 000000          N$ROA1  000022          SDBLH = ****** GX       T$NRUL  000007          $ERR67= ****** GX
LN.OOP= 000004          N$ROA2  000030          SDBNM = ****** GX       T$NVR   000001          $ERR68= ****** GX
LN.OPE= 000001          N$RTMX  000014          SDBS    002140R         T$RPRI  000040          $FLAGS= ****** GX
LN.REF= 000002          N$RTM1  000014          SDBSZ = ****** GX       T$SVC   000034          $FRKHD= ****** GX
LN.SER= 000002          N$RTM2  000015          SF.ACT= 000200          T$T5    000030          $HEADR= ****** GX
LN.STA= 000017          N$RT1   000000          SF.ENA= 000100          T$T6    000032          $MAPX = ****** GX
LN.SUB= 000360          N$RT2   000006          SF.LPB= 000004          T$$KMG= 000000          $PRCLN  000004RG    002
LN.TRI= 000006          N$TCTL  000112          SF.MFL= 000040          T$$MIN= 000000          $PREXT  000004RG    002
LTIM    000026R     002 N$TLC   000100          SF.PAC= 000020          UBMPR = 170200          $PRINC  000004RG    002
L$$ASG= 000000          N$TNC   000114          SF.REA= 000010          UISAR0= 177640          $PRIO = ****** GX
L$$DRV= 000000          N$TRC   000106          SF.SER= 000001          UISAR1= 177642          $PRLEN  000004RG    002
L$$P11= 000001          N$TTCB  000110          SF.SVC= 000002          VXBLK1  000156R     002 $PRLLN  000004RG    002
L$$11R= 000000          N$VER   000066          SF.UNL= 000040          VXBLK2  000170R     002 $RSXNM= ****** GX
L.COST  000015          N$XLEN  000124          SLTMA = ****** GX       VXBLK3  000202R     002 $SLTA = ****** GX
L.CTL   000012          N$$ACC= 000001          SLTNM = ****** GX       V$$CTR= 001000          $STKDP= ****** GX
L.CVA   177776          N$$BUF= 000001          SLTTOT  000310R     002 XAVL  = ****** GX       $SYMBL= ****** GX
L.DDM   000002          N$$LDV= 000001          STBERR  000544R         XBIAS = ****** GX       $XPAR = ****** GX
L.DDS   000004          N$$MCP= 000001          STBEXT  000020R     002 XBLK1   000120R     002 $$HTIM= ****** GX
L.DLC   000003          N$$MLL= 000001          SWR   = 177570          XBLK2   000132R     002 $$LTIM= ****** GX
L.DLM   000006          N$$MOV= 000010          SYSNM = ****** GX       XBLK3   000144R     002 .ALLOC= ****** GX
L.DLS   000010          N$$NCT= 000001          S$CEX   000000RG        XPTPDV  000034R     002 .AMAXC= ****** GX
L.FLG   000000          N$$PEM= 000001          S$$WRG= 000000          X$$DBT= 000000          .AMAXH= ****** GX
L.KRBA  000016          PAVL  = ****** GX       S$$YSZ= 007600          ZTIME = ****** GX       .BASEB= 140000
L.LEN = 000022          PDVID = ****** GX       S.COST  000001          Z.DAT = ****** GX       .CCBNM= ****** GX
L.MPF   000022          PDVNM = ****** GX       S.FLG   000000          Z.MAP = ****** GX       .CXALL= ****** GX
L.NMST  000020          PDVTA = ****** GX       S.LEN   000004          $ALL16= ****** GX       .CXCSR= ****** GX
L.NSTA  000014          PDVTOT  000322R     002 S.NMST  000002          $ALL18= ****** GX       .CXLEN= ****** GX
L.OWNR  000021          PIRQ  = 177772          S.OWNR  000003          $ALTLC  002314RG        .HOME = ****** GX
L.UNT   000013          PMODE = 030000          TKPS  = ****** GX       $AZFS = ****** GX       .LLCTB  000276R     002
MAXOV = ****** GX       PRO   = 000000          TK100 = ****** GX       $CLINS= ****** GX       .MAXC = ****** GX
MPAR  = 172100          PR1   = 000040          TPS   = 177564          $CTBIO= ****** GX       .MAXH = ****** GX
MPCSR = 177746          PR2   = 000100          TSKERR  000316R         $CXUIC= ****** GX       .MXPRO= ****** GX
M$$CRB= 000124          PR3   = 000140          T$FLAG  000044          $DEA16= ****** GX       .NA   = ****** GX
M$$CRX= 000000          PR4   = 000200          T$LIF   000013          $DIV  = ****** GX       .NBEA = ****** GX
M$$FCS= 000000          PR5   = 000240          T$LIFL  000013          $ERRQY= ****** GX       .NBRA = ****** GX
M$$MGE= 000000          PR6   = 000300          T$LIFO  000013          $ERRXQ= ****** GX       .NN   = ****** GX
M$$NET= 000000          PR7   = 000340          T$LIFS  000013          $ERRZA= ****** GX       .NOMAP= ****** GX
M$$OVR= 000000          PS    = 177776          T$LIN   000000          $ERR44= ****** GX       .NTPSZ= ****** GX
NBIAS = ****** GX       PUMR  = ****** GX       T$LIPS  000006          $ERR45= ****** GX       .PCB  = ****** GX
NETPF = ****** GX       P$$P45= 000000          T$LLD   000012          $ERR46= ****** GX       .PDVTB  000252R     002
NIPR    000030R     002 P$$WRD= 000000          T$LLDC  000045          $ERR47= ****** GX       .PUMR = ****** GX
NIRQ    000032R     002 P.BLKS= ****** GX       T$LLDL  000012          $ERR48= ****** GX       .RDBNM= ****** GX
NLBLK   000702R         P.REL = ****** GX       T$LLDO  000012          $ERR51= ****** GX       .ROUTB= ****** GX
NTLPT = ****** GX       QBIAS = ****** GX       T$LLDS  000012          $ERR53= ****** GX       .ROUTM= ****** GX
NTPT    000002R     002 QSTRT = ****** GX       T$LLEN  000046          $ERR54= ****** GX       .RPRI = ****** GX
N$ADJ1  000072          Q$$OPT= 000010          T$LOPR  000002          $ERR55= ****** GX       .SDBNM= ****** GX
```


Symbol table

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  ASL$,EMSG$R,GCL$,ISTAT$,NTLER$,STATE$,TRAN$,SLTDF$,DHBDF$

000000                                  SLTDF$                          ; DEFINE SLT OFFSETS
000000                                  DHBDF$                          ; DEFINE DECNET HOME BLOCK OFFSETS

                                ;
                                ; LOCAL MACROS
                                ;


                                ;
                                ; REJECT TPARS TRANSITION
                                ;
                                        .MACRO  REJ$
                                        ADD     #2,(SP)         ; RETURN TO CALLER+2
                                        CLR     SYNERR          ; INDICATE NO SYNTAX ERROR
                                        .ENDM   REJ$
```

```
                                        .SBTTL  CFLIN - ECHO LAST RECORD
                                        .SBTTL  FNLIN - ECHO FILE NAME AND LAST RECORD

                                ;+
                                ; CFLIN - ECHO THE LAST RECORD READ FROM THE CONFIG FILE
                                ; FNLIN - ECHO FILE NAME AND LAST RECORD READ FROM CONFIG FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
001312                          FNLIN:: CALL    $FNOUT          ; PRINT FILE NAME
001316  012700  000144'         CFLIN:: MOV     #CFGBF,R0       ; GET RECORD BUFFER ADDRESS
001322  016701  000352'                 MOV     CFGSZ,R1        ; AND RECORD SIZE
001326                                  CALLR   $TIOUT          ; PRINT IT ON TI:
```

```
                                        ;
                                        ; CHECK FOR LEGAL BLOCK SIZE VALUE
                                        ;
002050  005767  000000G       CHKBLK::TST     .PNUMH                  ; DOUBLE WORD VALUE?
002054  001012                       BNE     10$                     ; BR IF YES - ERROR
002056  016700  000000G              MOV     .PNUMB,R0               ; GET BLOCK SIZE VALUE
002062  022700  000040               CMP     #BLKSMN,R0              ; IS BLOCK SIZE WITHIN RANGE?
002066  101005                       BHI     10$                     ; BR IF NO - ERROR
002070  022700  002000               CMP     #BLKSMX,R0              ; ...
002074  103402                       BLO     10$                     ; BR IF NO - ERROR
002076  000241                       CLC                             ; INDICATE LEGAL VALUE
002100  000401                       BR      20$                     ; AND RETURN
002102  000261               10$:    SEC                             ; INDICATE ILLEGAL VALUE
002104                       20$:    RETURN
                                        ;
                                        ; CHECK FOR LEGAL WINDOW SIZE VALUE
                                        ;
002106  005767  000000G       CHKWND::TST     .PNUMH                  ; DOUBLE WORD VALUE?
002112  001010                       BNE     10$                     ; BR IF YES - ERROR
002114  016700  000000G              MOV     .PNUMB,R0               ; GET WINDOW BLOCK VALUE
002120  002405                       BLT     10$                     ; BR IF ILLEGAL VALUE
002122  022700  000177               CMP     #WNDSMX,R0              ; WITHIN VALID RANGE?
002126  103402                       BLO     10$                     ; BR IF NO - ERROR
002130  000241                       CLC                             ; INDICATE LEGAL VALUE
002132  000401                       BR      20$                     ; AND RETURN
002134  000261               10$:    SEC                             ; INDICATE ILLEGAL VALUE
002136                       20$:    RETURN
000001                               .END
```

```
                                        .SBTTL  LOCAL DATA
                                ;
                                ; LOCAL DATA
                                ;
000000                                  .PSECT  DATA,D
                                        .NLIST  BEX
                                ;
                                ; CONFIG FILE NAME AND EXTENSION
                                ;
000000  103     105     124     CFGNAM: .ASCIZ  "CETAB"
000006  115     101     103     CFGEXT: .ASCIZ  "MAC"
                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
000012  015                     ERRMSG: .BYTE   15
000013  116     124     114             .ASCII  "NTL -- "
000022                          ERRBUF: .BLKB   70.

                                        .EVEN
```

```
                                        .SBTTL  REP.NN - REPLACE SIGNED DECIMAL NUMBER

                                ;+
                                ;
                                ; REP.NN - REPLACE SIGNED DECIMAL NUMBER
                                ;
                                ; INPUTS:
                                ;       R5 = ERROR MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       RDB SIZE STORE IN BUFFER
                                ;
                                ;-

001136                          REP.NN::
001136  010500                          MOV     R5,R0                   ; COPY ERROR MESSAGE BLOCK ADDRESS
001140  005720                          TST     (R0)+                   ; POINT TO OUTPUT BUFFER
001142  122720  000052          10$:    CMPB    #'*,(R0)+               ; SEARCH FOR ASTERISK
001146  001375                          BNE     10$                     ; ...
001150  005300                          DEC     R0                      ; BACK UP TO ASTERISK
001152  016701  000000G                 MOV     .RDBSZ,R1               ; GET RDB SIZE
001156  005002                          CLR     R2                      ; ZERO SUPPRESSION
001160                                  CALL    $CBDSG                  ; CONVERT TO DECIMAL SIGNED
001164  112720  000056                  MOVB    #'.,(R0)+               ; INSERT DECIMAL INDICATOR (PERIOD)
001170  105020                          CLRB    (R0)+                   ; SET END OF STRING
001172                                  RETURN
```

1103

NTL
SYM
SYM
FNB
FNL
F.N
F.U
F.V
HEX
IE.
IE.
L8R
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LF.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LN.
LO.
LO.
LO.
LO.
LO.
LO.
LO.
LO.
LO.
LR.
LR.
LR.
LR
LR.

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
FNBUFF	002166 RG	#7-218 7-219 7-220 7-221
FNLIN	001046 RG	7-109 #13-589
F.NXT	000000	#5-72
F.USE	000002	#5-72
F.VAR	000003	#5-72
HEXSTR	002064 RG	#7-199
IE.EOF	= ****** GX	10-469
IE.RBG	= ****** GX	10-472
LBRA	= 000074	#7-263
LF.ACT	= 100000	#5-71
LF.BRO	= 000400	#5-71
LF.BWT	= 000007	#5-71
LF.ENA	= 002000	#5-71 8-322
LF.LPB	= 001000	#5-71
LF.MDC	= 000100	#5-71
LF.MFL	= 004000	#5-71 8-322
LF.MTP	= 000020	#5-71
LF.PAC	= 000200	#5-71
LF.RDY	= 040000	#5-71
LF.REA	= 010000	#5-71
LF.SER	= 000040	#5-71 8-322
LF.TIM	= 000010	#5-71
LF.UNL	= 020000	#5-71
LF.X2P	= 000000	#5-71
LN.CLO	= 000000	#5-71
LN.DUM	= 000005	#5-71
LN.LOA	= 000004	#5-71
LN.LOO	= 000003	#5-71
LN.OAU	= 000003	#5-71
LN.OFF	= 000001	#5-71
LN.ON	= 000000	#5-71
LN.OOP	= 000004	#5-71
LN.OPE	= 000001	#5-71
LN.REF	= 000002	#5-71
LN.SER	= 000002	#5-71
LN.STA	= 000017	#5-71
LN.SUB	= 000360	#5-71
LN.TRI	= 000006	#5-71
LO.CON	000006	#5-69
LO.CSR	000006	#5-69
LO.CTL	000014	#5-69
LO.LIN	000004	#5-69
LO.PAR	000010	#5-69
LO.PRI	000010	#5-69
LO.SCR	000014	#5-69
LO.STA	000015	#5-69
LO.VCT	000012	#5-69
LR.AST	000026	#5-69
LR.CIR	000002	#5-69
LR.CTL	000004	#5-69
LR.EFN	000004	#5-69
LR.LIN	000002	#5-69

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 3

```
                                        .SBTTL  LOCAL DATA


                                ;
                                ; LOCAL DATA
                                ;

000000                                  .PSECT  DATA,D

                                        .NLIST  BEX


                                ;
                                ; CONFIG FILE NAME AND EXTENSION
                                ;
000000  103     105     124     CFGNAM: .ASCIZ  "CETAB"
000006  115     101     103     CFGEXT: .ASCIZ  "MAC"


                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
000012  015                     ERRMSG: .BYTE   15
000013  116     124     114             .ASCII  "NTL -- "
000022                          ERRBUF: .BLKB   70.

                                        .EVEN

                                ;
                                ; RECORD BUFFER AND BUFFER LENGTH
                                ;
000130                          CFGBF:: .BLKB   134.
000336                          CFGSZ:: .BLKW   1

000340                          SPSAV:  .BLKW   1               ; SAVED STACK POINTER FOR ERROR EXIT
000342                          SYNERR::.BLKW   1               ; SYNTAX ERROR FLAG
000344                          MRKFL:: .BLKW   1               ; MARKED-FOR-LOAD FLAG

000346                          LINNAM::.BLKW   1               ; DEVICE NAME FROM LINE-ID (RAD50)
000350                          LINCTL::.BLKB   1               ; CONTROLLER NUMBER
000351                          LINUNT::.BLKB   1               ; UNIT NUMBER
000352                          NEXT::  .BLKW   1               ; ADDRESS OF NEXT AVAILABLE BYTE
000354                          PORTNO::.BLKW   1               ; PORT NUMBER FROM PSN$DF MACRO
000356                          COUNT:: .BLKB   1               ; COUNT OF NUMBER OF DIGITS IN DTE ADDRESS
000357                          CVTBUF::.BLKB   1               ; CONVERSION BUFFER FOR ASCII TO DECIMAL DIGIT
                                        .EVEN

                                ;
                                ; DATA FOR X3P$DF MACRO
                                ;
000360                          BLKSZD::.BLKW   1               ; DEFAULT BLOCK SIZE
000362                          BLKSZM::.BLKW   1               ; MAXIMUM BLOCK SIZE
000364                          WNDSZD::.BLKW   1               ; DEFAULT WINDOW SIZE
000366                          WNDSZM::.BLKW   1               ; MAXIMUM WINDOW SIZE

                                ;
                                ; DATA FOR DTE$DF MACRO
                                ;
000370                          DTEDES::.BLKW   1               ; DTE ADDRESS
000372                          DTEFLG::.BLKW   1               ; FLAGS WORD
        000001                          DE.ON   == 1            ; SET DTE STATE TO "ON"
        000002                          DE.OFF  == 2            ; SET DTE STATE TO "OFF"
                                        .EVEN
```

```
                                        .SBTTL  FNDPDV - FIND PDV
                                ;+
                                ; FNDPDV - TRY TO FIND PDV
                                ;
                                ; INPUTS:
                                ;       R0=RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=PDV ADDRESS
                                ;       R1=PDV INDEX
                                ;       R2=DESTROYED
                                ;-
001012                          FNDPDV::
001012  010046                          MOV     R0,-(SP)        ;; SAVE NAME TO BE MATCHED
001014  017701  000000G                 MOV     @PDVTA,R1       ;; GET PDV TABLE ADDRESS
001020  017702  000000G                 MOV     @PDVNM,R2       ;; AND NUMBER OF PDV'S
001024  001407                          BEQ     30$             ;; IF ZERO, NO PDV'S!
001026  012100                  10$:    MOV     (R1)+,R0        ;; GET PDV ADDRESS
001030  001403                          BEQ     20$             ;; IF ZERO, SKIP THIS ONE
001032  026016  000000G                 CMP     Z.NAM(R0),(SP)  ;; DOES THE NAME MATCH?
001036  001403                          BEQ     40$             ;; IF EQ, YES (C-BIT CLEAR)
001040                          20$:    SOB     R2,10$          ;; LOOP $PDVNM TIMES
001044  000261                  30$:    SEC                     ;; INDICATE FAILURE
001046  005226                  40$:    INC     (SP)+           ;; PURGE STACK
001050  006016                          ROR     (SP)            ;; SAVE C-BIT
001052  005741                          TST     -(R1)           ;; CALCULATE PDV INDEX
001054  167701  000000G                 SUB     @PDVTA,R1       ;;      ...
001060  006316                          ASL     (SP)            ;; RESTORE C-BIT
001062                                  RETURN
```

```
                        ;+
                        ; $NLCSR - TEST FOR DEVICE ONLINE
                        ;
                        ; THIS ROUTINE IS CALLED TO TEST THE DEVICE CSR FOR ONLINE/OFFLINE. IF ODT
                        ; IS BEING USED WITH NTL, THE ODD ADDRESS TRAP IS SERVICED BY ODT, NOT BY
                        ; THE ROUTINE "OFFLIN".
                        ;
                        ; INPUTS:
                        ;       R0 = CSR ADDRESS
                        ;       $URM = UNIBUS RUN MASK FOR MULTI-PROCESSOR RSX-11M+ SYSTEMS
                        ;
                        ; OUTPUTS:
                        ;       C-BIT = SUCCESS/FAILURE
                        ;-
000000                  $NLCSR::
000000  005046                  CLR     -(SP)           ; ASSUME SUCCESS

                                .IF DF  R$$MPL

                                TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
                                BEQ     20$             ; IF EQ, NO
                                MOV     $URM,SETAF+S.AFAF ; ELSE, RUN ON THE RIGHT PROCESSOR
                                DIR$    #SETAF          ; SET AFFINITY
                                BCS     10$             ; IF CS, CAN'T
                                TST     $DSW            ; WAS IT SUCCESSFUL ?
                                BPL     20$             ; IF PL, YES
                        10$:    INC     (SP)            ; INDICATE FAILURE
                                BR      30$             ; AND LEAVE
                        20$:

                                .ENDC

000002                          DIR$    #SST            ; SETUP SST TO CATCH NXM AND MP ONLY

000010                          SWSTK$  22$             ; ENTER SYSTEM STATE
000014  052766  000000G 000020          BIS     #PR7,20(SP)     ; RETURN TO USER STATE AT PRIORITY 7
000022  017767  000000G 000014'         MOV     @USAR7,UAR7SV   ; SAVE USER APR7
000030  017767  000000G 000016'         MOV     @USDR7,UDR7SV   ; SAVE USER PDR7
000036  017777  000000G 000000G         MOV     @KSAR7,@USAR7   ; MAP USER TASK TO I/O PAGE
000044  052777  077400  000000G         BIS     #77400,@USDR7   ; ACCESS 4KW

000052                          RETURN

000054  005710                 22$:    TST     (R0)            ; TRY TO ACCESS DEVICE REGISTER
000056                          SWSTK$  25$             ; ENTER SYSTEM STATE
000062  016777  000016' 000000G         MOV     UDR7SV,@USDR7   ; RESTORE USER PDR7
000070  016777  000014' 000000G         MOV     UAR7SV,@USAR7   ; RESTORE USER APR7
000076  042766  000000G 000020          BIC     #PR7,20(SP)     ; RETURN TO USER STATE AT PRIORITY 0
000104                          RETURN

000106                          25$:    DIR$    #NOSST          ; TURN OFF SST'S COMPLETELY

                                .IF DF  R$$MPL

                                TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
                                BEQ     30$             ; IF EQ, NO
                                RMAF$S                  ; REMOVE THE CPU AFFINITY
```

```
                                        .SBTTL  $NLDEV - SET UP DEVICE DATA STRUCTURES

                                ;+
                                ; $NLDEV - SETUP DEVICE DATA STRUCTURES
                                ;
                                ; THIS ROUTINE IS CALLED WHEN DOING A "SET PROCESS" OF AN LLC IF THE LLC$DF
                                ; MACRO IN CETAB INCLUDED A DEVICE NAME (SIXTH MACRO ARGUMENT). IF THE DEVICE
                                ; DATA STRUCTURES ALREADY EXIST, THE PCB ADDRESS IS COPIED FROM THE PDV INTO
                                ; THE DCB. IF THE DATA STRUCTURES DO NOT EXIST, THE FILE [131,54]XXXTAB.TSK
                                ; IS ASSUMED TO BE AN IMAGE COPY OF THE DEVICE TABLES.
                                ;
                                ; INPUTS:
                                ;       $$DEV=2 CHARACTER DEVICE NAME (ASCII)
                                ;       $PDVA=PDV ADDRESS
                                ;       $NAME=PROCESS NAME (RAD50)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000000                          $NLDEV::
                                        .IFT    ; DF R$$11M

000000                                  CALL    FIND            ; SEE IF DATA STRUCTURES ALREADY EXIST
000004  103440                         BCS     20$             ; IF CS, DRIVER ALREADY RESIDENT!
000006                                  CALL    DVSYM           ; LOOK FOR SPECIAL SYMBOLS
000012  103435                         BCS     20$             ; IF CS, STB FILE ERROR
000014  005767  000166'                 TST     FOUND           ; DO STRUCTURES ALREADY EXIST?
000020  001013                         BNE     10$             ; IF NE, YES
000022                                  CALL    DVLBL           ; OPEN IMAGE FILE, READ LABEL BLOCK
000026  103427                         BCS     20$             ; IF CS, FAILURE
000030                                  CALL    DVALL           ; ALLOCATE SPACE FROM THE POOL
000034  103424                         BCS     20$             ; IF CS, FAILURE
000036                                  CALL    DVXFR           ; TRANSFER STRUCTURES INTO CORE
000042  103421                         BCS     20$             ; IF CS, FAILURE
000044                                  CALL    $CLDEQ          ; CLOSE IMAGE FILE
000050  016703  000150'          10$:   MOV     DCB,R3          ; GET DCB ADDRESS
000054  016763  000000G 000000G         MOV     $$TBL,D.DSP(R3) ; COPY DISPATCH TABLE ADDRESS
000062  001412                         BEQ     10$             ; IF EQ, ERROR
000064  016763  000000G 000000G         MOV     $PCB,D.PCB(R3)  ; COPY PCB ADDRESS
000072  016704  000172'                 MOV     UNITS,R4        ; GET NUMBER OF UNITS
000076  016305  000000G                 MOV     D.UCB(R3),R5    ; GET FIRST UCB ADDRESS

                                        .IF DF  R$$MPI
                                        SAVRG   <R5>            ; SAVE UCB ADDRESS
                                        MOV     U.SCB(R5),R5    ; COPY SCB ADDRESS
                                        ADD     SKS5,R5         ; POINT TO S.KS5 OFFSET
                                        MOV     $BIAS,(R5)      ; SETUP S.KS5 FOR HRC
                                        RESRG   <R5>            ; RESTORE UCB ADDRESS

                                        .ENDC

000102                                  CALL    $NLCDD          ; CALL DRIVER AT POWERFAIL ENTRY
000106                          20$:    RETURN

                                ;
```

```
                                        .SBTTL  REP4A - REPLACE XXTAB NAME AND SYMBOL NAME
                                        .SBTTL  REP4A1 - REPLACE XXX NAME AND SYMBOL NAME

                                ;+
                                ;
                                ; REP4A - REPLACE XXXTAB NAME AND SYMBOL NAME
                                ; REP4A1 - REPLACE XXX NAME AND SYMBOL NAME
                                ;
                                ; INPUTS:
                                ;       R1=SYMBOL LOOKUP CONTROL BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R1=DESTROYED
                                ;-
001446  010146          REP4A:  MOV     R1,-(SP)        ; SAVE R1
001450                          CALL    $REP.P          ; REPLACE XXXTAB NAME
001454  000420                 BR      REP4A2
001456  010146          REP4A1: MOV     R1,-(SP)        ; SAVE R1
001460  112314          10$:    MOVB    (R3)+,(R4)      ; COPY FORMAT STRING
001462  122724  000052          CMPB    #'*,(R4)+       ; LOOKING FOR FIRST ASTERISK
001466  001374                 BNE     10$             ;       ...
001470  005304                 DEC     R4              ; SKIP THE ASTERISK
001472  010400                 MOV     R4,R0           ; COPY BUFFER ADDRESS
001474  016701  000006G        MOV     $NAME,R1        ; GET LLC NAME
001500                          CALL    $C5TA           ; CONVERT TO ASCII
001504  122740  000040  20$:    CMPB    #' ,-(R0)       ; DISCARD TRAILING SPACES
001510  001775                 BEQ     20$             ;       ...
001512  005200                 INC     R0              ;       ...
001514  010004                 MOV     R0,R4           ; SET NEW BUFFER ADDRESS
001516  012601          REP4A2: MOV     (SP)+,R1        ; RESTORE R1
001520  112314          10$:    MOVB    (R3)+,(R4)      ; COPY FORMAT STRING
001522  122724  000052          CMPB    #'*,(R4)+       ; LOOKING FOR THE SECOND ASTERISK
001526  001374                 BNE     10$             ;       ...
001530  010400                 MOV     R4,R0           ; COPY BUFFER POINTER
001532  112744  000042          MOVB    #'",-(R4)       ; DEPOSIT A DOUBLE QUOTE
001536  016146  000006          MOV     6(R1),-(SP)     ; SAVE SECOND HALF OF NAME
001542  016101  000004          MOV     4(R1),R1        ; GET FIRST HALF OF NAME
001546                          CALL    $C5TA           ; CONVERT TO ASCII
001552  012601                 MOV     (SP)+,R1        ; RESTORE SECOND HALF OF NAME
001554                          CALL    $C5TA           ; CONVERT TO ASCII
001560  122740  000040  20$:    CMPB    #' ,-(R0)       ; DISCARD TRAILING BLANKS
001564  001775                 BEQ     20$             ;       ...
001566  005200                 INC     R0              ;       ...
001570  112720  000042          MOVB    #'",(R0)+       ; DEPOSIT ANOTHER DOUBLE QUOTE
001574  010004                 MOV     R0,R4           ; SET NEW BUFFER POINTER
001576                          RETURN
```

MAPX CREATED BY MACRO ON 30-JUN-85 AT 00:00 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
APR	000000	RG	#4-53	*5-90	5-94
APRSV	000002	RG	#4-57	*5-91	5-94
ISAR5	= ******	GX	5-80	6-135	6-136
R$$PRO	= ******		5-78	6-117	6-133
S$$BAS	= ******		5-78	6-117	6-133
XBIAS	= ******	GX	6-135		
$MAPX	000000	RG	#5-77		
$MAPXP	000034	RG	#6-111		

MAPX CREATED BY MACRO ON 30-JUN-85 AT 00:00 AGE 2

```
                        .TITLE  NTLCEX - LOAD COMM EXEC IMAGE
                        .IDENT  /V05.00/

;
;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - "SET SYS" ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DEcnet-Micro/RSX V1.0
;
;
```

```
    639 001266  060120                  6$:     ADD     R1,(R0)+        ; RELOCATE TABLE
    640 001270                                   SOB     R2,6$           ;  ...
    641 001274  017700  000000G                  MOV     @LLCTA,R0       ; RELOCATE THE REVERSE MAPPING TABLE
    642 001300  017702  000000G                  MOV     @SLTNM,R2       ; SAME NUMBER OF ENTRIES AS IN SLT TABLE
    643 001304  006201                           ASR     R1              ; DIVIDE CETAB START ADDRESS BY 2
    644 001306  005710                   7$:     TST     (R0)            ; IS THIS A STATION TABLE ADDRESS ?
    645 001310  100402                           BMI     8$              ; IF MI, YES .. RELOCATE
    646 001312  005720                           TST     (R0)+           ; ELSE, SKIP IT
    647 001314  000403                           BR      9$              ; LOOK FOR MORE
    648 001316  060110                   8$:     ADD     R1,(R0)         ; RELOCATE ADDRESS
    649 001320  052720  100000                   BIS     #100000,(R0)+   ; MAKE SURE HI BIT SET
    650 001324                           9$:     SOB     R2,7$           ; REPEAT <$SLTNM> TIMES
    651
    652                                          .IF DF  R$$MPL
    653
    654                                          CALL    CECLK           ; SET UP MULTI-PROCESSOR CLOCK QUEUE ENTRIES
    655
    656                                          .ENDC
    657
    658 001330  032767  000000G 000000G 10$:     BIT     #FL.NTI,$FLAGS  ; IS THIS SET SYS + NTINIT ?
    659 001336  001402                           BEQ     20$             ; IF EQ, NO
    660 001340                                   CALL    CENTI           ; ELSE GO DO NTINIT WORK ALSO
    661 001344                           20$:    RETURN
    662
    663                                  ;
    664                                  ; ERROR CONDITION
    665                                  ;
    666 001346                           101$:   EMSG$R  ZA              ; INCONSISTENT COMM EXEC IMAGE
```

```
                                                                L 4
NTLCEX - LOAD COMM EXEC IMAGE   MACRO V05.03b Saturday 29-Jun-85 00:22  Page 18-4
Symbol table

.SLTMB  000264R     002 .UMRFL= ****** GX       .$$$$.= 000034

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        002666    001   (RW,I,LCL,REL,CON)
DATA    000410    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 63
Work  file writes: 64
Size of work file: 20486 Words  ( 81 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:35.28
SY:NTLCEX.V2,[132,134]NTLCEX/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLCEX
```

```
                                                                M 4
NTLCEX     CREATED BY  MACRO  ON 29-JUN-85 AT 00:23     PAGE 1
```

```
                                .SBTTL  CONSTANT DEFINITIONS

                        ;
                        ; CONSTANTS
                        ;
000007          MAXWND = 7              ; MAXIMUM WINDOW SIZE (SVC$DF)
000200          MAXNDC = 128.           ; MAXIMUM NUMBER OF NODE COUNTERS (LLC$DF)
000040          BLKSMN = 32.            ; MINIMUM BLOCK SIZE VALUE
002000          BLKSMX = 1024.          ; MAXIMUM BLOCK SIZE VALUE
000177          WNDSMX = 127.           ; MAXIMUM WINDOW SIZE VALUE
007777          CHNLMX == 4095.         ; MAXIMUM VALUE FOR CHANNEL NUMBER
```

```
                                        .SBTTL  REP8A - REPLACE DEVICE NAME AND CONTROLLER NUMBER

                                ;+
                                ; REP8A - REPLACE DEVICE NAME AND CONTROLLER NUMBER
                                ;
                                ; INPUTS:
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                ;       NAME=LINE (DDM) NAME
                                ;       CTL=CONTROLLER NUMBER
                                ;
                                ; OUTPUTS:
                                ;       R0=DESTROYED
                                ;       R3,R4=UPDATED
                                ;-
001332  112314                 REP8A:: MOVB    (R3)+,(R4)      ; LOOK FOR THE ASTERISK
001334  122724  000052                  CMPB    #'*,(R4)+       ;       ...
001340  001374                          BNE     REP8A           ;       ...
001342  012700  000012'                 MOV     #NAME,R0        ; POINT AT LINE ID INFO
001346  112064  177777                  MOVB    (R0)+,-1(R4)    ; COPY LINE NAME
001352  112024                          MOVB    (R0)+,(R4)+     ;       ...
001354  122720  000040                  CMPB    #' ,(R0)+       ; SKIP TRAILING BLANK (IF ANY)
001360  001402                          BEQ     10$             ;       ...
001362  116024  177777                  MOVB    -1(R0),(R4)+    ; COPY LAST CHARACTER OF NAME
001366  112724  000055         10$:     MOVB    #'-,(R4)+       ; DEPOSIT A HYPHEN
001372  111014                          MOVB    (R0),(R4)       ; COPY CONTROLLER NUMBER
001374  152724  000060                  BISB    #'0,(R4)+       ; MAKE IT AN ASCII NUMBER
001400                                  RETURN
```

Symbol table

```
ADDDF   000100R      003 D$NBEA  000056          LF.MFL= 004000          MS.MTP= ****** GX       T$$KMG= 000000
AD.NUM  001534R          D$NBRA  000054          LF.MTP= 000020          MS.MUX= ****** GX       T$$MIN= 000000
A$$CHK= 000000           D$NEND= 000054          LF.PAC= 000200          M$$CRB= 000124          UCNT    000142RG    002
A$$CPS= 000000           D$NLN   000030          LF.RDY= 040000          M$$CRX= 000000          UNT     000016RG    002
A$$PRI= 000000           D$NN    000060          LF.REA= 010000          M$$FCS= 000000          V$$CTR= 001000
A$$TRP= 000000           D$OUTT  000043          LF.SER= 000040          M$$MGE= 000000          WNDSMX= 000177
BITS    000000R      003 D$RETF  000050          LF.TIM= 000010          M$$NET= 000000          X$$DBT= 000000
BITS1   000030R      003 D$RNN   000002          LF.UNL= 020000          M$$DVR= 000000          ZF.PSE= ****** GX
BITS2   000044R      003 D$RTMR  000076          LF.X2P= 000000          NAME    000012RG    002 Z.FLG = ****** GX
BLKSMN= 000040           D$SEG   000036          LLCDF   000114R     003 N$$ACC= 000001          Z.NAM = ****** GX
BLKSMX= 002000           D$SER   000032          LLCDF1  000152R     003 N$$BUF= 000001          $ALPHA= 000022
CERR    001214RG         D$SQRL  000052          LNKEND  002016RG        N$$LDV= 000001          $ANY  = 000020
CFERR   001224RG         D$$BUG= 177514          LN.CLO= 000000          N$$MCP= 000001          $AZFS = ****** GX
CFGBF   000144RG     002 D$$ISK= 000000          LN.DUM= 000005          N$$MLL= 000001          $BLANK= 000006
CFGEXT  000006R      002 D$$L11= 000001          LN.LOA= 000004          N$$MDV= 000010          $CEACX= ****** GX
CFGFLG  000136RG     002 D$$YNC= 000000          LN.LOD= 000003          N$$NCT= 000001          $CFUIC= ****** GX
CFGKW   000000RG     004 D$$YNM= 000000          LN.DAU= 000003          N$$PEM= 000001          $CLBUF= ****** GX
CFGNAM  000000R      002 D.NAME  001402RG        LN.OFF= 000001          OPEN    000772R         $CLDEQ= ****** GX
CFGST   000000RG     003 END     000070R     003 LN.ON = 000000          PDVNM = ****** GX       $CLIOS= ****** GX
CFGSZ   000352RG     002 ERRBUF  000027RG     002 LN.OOP= 000004          PDVTA = ****** GX       $CLQUE= ****** GX
CFLIN   001316RG         ERRMSG  000017RG     002 LN.OPE= 000001          PLTDF   000244R     003 $CLSIZ= ****** GX
CHKBLK  002050RG         EXT     000206R     003 LN.REF= 000002          PL.CHN  001566R         $CTIM = ****** GX
CHKWND  002106RG         E$$XPR= 000000          LN.SER= 000002          P$$P45= 000000          $C5TA = ****** GX
CHNLMX= 007777 G         FG.X2P= 000001 G        LN.STA= 000017          P$$WRD= 000000          $DIGIT= 000024
CTL     000015RG     002 FG.X3P= 000002 G        LN.SUB= 000360          P.EXT   001506R         $DNUMB= 000014
C$$CKP= 000000           FMT8    001626RG     002 LN.TRI= 000006          P.INC   001466RG        $EOS  = 000012
C$$ORE= 000400           FMT8A   001646RG     002 L$$ASG= 000000          QADD    001134R         $ERRYP  001356RG    002
C$$RSH= 177564           FMT8B   001677RG     002 L$$DRV= 000000          QLLC    001142R         $ERRYR  001400RG    002
DECPT = ****** GX        FM.8  = 000000          L$$P11= 000001          QPLT    001126R         $ERRZP  001434RG    002
DE.OFF= 000002 G         FM.8B = 000000          L$$11R= 000000          Q$$OPT= 000010          $ERRZQ  001304RG    002
DE.ON = 000001 G         FNDPDV  001744RG        L.CH1   001664R         READ    001034R         $ERR1A  001062R     002
DTEDES  000360RG     002 FNLIN   001312RG        L.CH2   001674R         REP8A   001332RG        $ERR1P  000370R     002
DTEFLG  000362RG     002 F$$LVL= 000001          L.COST  000015          RTSPC   001310RG        $ERR1Q  000426R     002
D$AMXC  000072           GETCSR  001622RG        L.CTIM  001704R         R$$DER= 000000          $ERR1R  000464R     002
D$AMXH  000074           G$$TPP= 000000          L.CTL   000012          R$$K11= 000001          $ERR1S  000512R     002
D$ANN   000000           G$$TSS= 000000          L.CVA   177776          R$$SND= 000000          $ERR1T  000542RG    002
D$BRPR  000102           G$$TTK= 000000          L.DDM   000002          R$$11M= 000000          $ERR1U  000570R     002
D$BRTM  000100           G$$WRD= 000000          L.DDS   000004          SCNT    000140RG    002 $ERR1V  000626R     002
D$DELF  000045           HSHADD  000366RG     002 L.DLC   000003          SF.ACT= 000200          $ERR1W  000664R     002
D$DELW  000046           HSHSZ   000364RG     002 L.DLM   000006          SF.ENA= 000100          $ERR1X  000722R     002
D$END = 000104           IE.EOF= ****** GX        L.DLS   000010          SF.LPB= 000004          $ERR1Y  000760R     002
D$FNB   000034           IE.RBG= ****** GX        L.FLG   000000          SF.MFL= 000040          $ERR1Z  001024R     002
D$HIOR  000024           INC     000176R     003 L.KRBA  000016          SF.PAC= 000020          $ERR2A  001124RG    002
D$HOST  000022           I$$RAR= 000000          L.LEN = 000022          SF.REA= 000010          $ERR2C  001174RG    002
D$INAC  000044           I$$RDN= 000000          L.MPF   000022          SF.SER= 000001          $ERR2F  001230RG    002
D$INCT  000042           K$$CNT= 177546          L.NAME  001402R         SF.SVC= 000002          $ERR2I  001264RG    002
D$IPL   000051           K$$CSR= 177546          L.NMST  000020          SF.UNL= 000040          $ERR3A  001466R     002
D$LID   000020           K$$LDC= 000000          L.NSTA  000014          SPSAV   000354R     002 $ERR3B  001524R     002
D$LNAM  000006           K$$TPS= 000074          L.OWNR  000021          SYNERR  000356RG    002 $ERR3C  001562R     002
D$LNUM  000014           LCTIM   000236R     003 L.SEG   001730R         S$$WRG= 000000          $EXIT = 000000
D$LST   000047           LD$LP = 000000          L.UNT   000013          S$$YSZ= 007600          $FAIL = 177777
D$MAXC  000064           LF.ACT= 100000          MAXNDC= 000200          S.COST  000001          $FEATR= ****** GX
D$MAXH  000066           LF.BRD= 000400          MAXWND= 000007          S.FLG   000000          $FNOUT= ****** GX
D$MAXV  000070           LF.BWT= 000007          MS.CIR= ****** GX       S.LEN   000004          $GCLEP= ****** GX
D$MLL   000040           LF.ENA= 002000          MS.EXT= ****** GX       S.NAME  001402RG        $LAMDA= 000000
D$MNOD  000041           LF.LPB= 001000          MS.INC= ****** GX       S.NMST  000002          $LINKX= ****** GX
D$NA    000062           LF.MDC= 000100          MS.LLC= ****** GX       S.OWNR  000003          $MISS = ****** GX
```


Symbol table

```
                                        .SBTTL  ERROR MESSAGES


                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000130                                  NTLER$  <:>,1P,8,CFERR,$REP.N,$FNOUT,<Open Failure (-***.)>
000166                                  NTLER$  <:>,1Q,8,CFERR,$REP.N,$FNOUT,<Read Failure (-***.)>
000224                                  NTLER$  <:>,1R,8,CFERR,RTSPC,FNLIN,<Read Overflow>
000252                                  NTLER$  <:>,1T,8,CERR,RTSPC,CFLIN,<Syntax Error>
                                        .IF DF  R$$11M
000300                                  NTLER$  <:>,XA,8B,CERR,REP8B,CFLIN,<Not Common>
000324                                  NTLER$  <:>,XB,8B,CERR,REP8B,CFLIN,<Busy>
000342                                  NTLER$  <:>,1Z,9,CWARN,REPND,,<System name changed to "******">

                                        .IFTF
000412                                  NTLER$  <:>,XC,8B,CERR,REP8B,CFLIN,<Not In System>
                                        .IFT
000440                                  NTLER$  <:>,XD,8B,CERR,REP8B,CFLIN,<Too Small For RDBS>
                                                .IF DF  M$$MGE
000474                                  NTLER$  <:>,XE,8B,CERR,REP8B,CFLIN,<Not System-Controlled>
000532                                  NTLER$  <:>,XF,8,CERR,RTSPC,CFLIN,<NonUMR Area Too Large>
                                                .ENDC
                                        .IFF
                                        NTLER$  <:>,XF,8,CERR,RTSPC,CFLIN,<Non-UMR-Mapped Area Too Large>
                                        .IFTF
000570                                  NTLER$  <:>,XG,8,CFERR,RTSPC,,<NO BUF$DF ?>
000614                                  NTLER$  <:>,XH,8,CFERR,RTSPC,,<NO PAR$DF ?>
000640                                  NTLER$  <:>,XI,8,CERR,RTSPC,CFLIN,<NO BUF$DF ?>
                                        .IFT
                                                .IF DF  M$$MGE
000664                                  NTLER$  <:>,XJ,8,CERR,RTSPC,CFLIN,<Sub-PCB Alloc. Failure>
000724                                  NTLER$  <:>,XK,8B,CERR,REP8B,CFLIN,<Too Fragmented>
000754                                  NTLER$  <:>,XL,8,CERR,RTSPC,CFLIN,<UMR Blk. Alloc. Failure>
001014                                  NTLER$  <:>,XM,8,CERR,RTSPC,CFLIN,<Not Enough Contig. UMRs>
                                                .ENDC
                                        .IFF
                                        NTLER$  <:>,XJ,8C,CERR,REP8C,CFLIN,<Creation Failed>
                                        NTLER$  <:>,XK,8C,CERR,REP8C,CFLIN,<Mapping Failed>
                                        .ENDC
001054                                  NTLER$  <:>,XV,8,CERR,RTSPC,,<Binary buffer overflow>
001114                                  NTLER$  <:>,XO,8,CERR,RTSPC,CFLIN,<Illegal max logical links value>
001164                                  NTLER$  <:>,XP,8,CERR,RTSPC,CFLIN,<Illegal max node counter value>
001234                                  NTLER$  <:>,XQ,8,CERR,RTSPC,CFLIN,<Illegal delay factor value>
001300                                  NTLER$  <:>,XR,8,CERR,RTSPC,CFLIN,<Illegal delay weight value>
001344                                  NTLER$  <:>,XS,8,CERR,RTSPC,CFLIN,<Illegal packet limiter value>
001412                                  NTLER$  <:>,XT,8,CERR,RTSPC,CFLIN,<Illegal link service threshold value>
001470                                  NTLER$  <:>,XU,8,CERR,RTSPC,CFLIN,<Illegal incoming timer value>
001536                                  NTLER$  <:>,XY,8,CERR,RTSPC,CFLIN,<Illegal outgoing timer value>
001604                                  NTLER$  <:>,XW,8,CERR,RTSPC,CFLIN,<Illegal inactivity timer>
001646                                  NTLER$  <:>,XX,8,CERR,RTSPC,CFLIN,<Illegal retransmit value>

                                        .DSABL  LC


                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
```

```
                                        .SBTTL  REPND - REPLACE NODE NAME

                                ;+
                                ;
                                ; REPND - PLACE NODE NAME IN ERROR MESSAGE TEXT
                                ;
                                ; INPUTS:
                                ;       R5 - ERROR MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2 DESTROYED
                                ;       NEW NODE NAME IN BUFFER
                                ;
                                ;-

                                        .IF     DF,R$$1'M
001174  010500                  REPND:: MOV     R5,R0           ; COPY ERROR MESSAGE BLOCK ADDRESS
001176  005720                          TST     (R0)+           ; POINT TO OUTPUT BUFFER
001200  122720  000052          5$:     CMPB    #'*,(R0)+       ; LOOK FOR ASTERISK
001204  001375                          BNE     5$              ; BR IF NOT ASTERISK
001206  005300                          DEC     R0              ; BACK UP TO ASTERISK
001210  012702  000006                  MOV     #6,R2           ; NODE NAME IS MAXIMUM OF 6 CHARACTERS
001214  017701  000000G                 MOV     @DECPT,R1       ; STORE DECNET HOME BLOCK
001220  062701  000006                  ADD     #D$LNAM,R1      ; GET ADDRESS OF NODE NAME TO COPY (ASCII)
001224  112110                  10$:    MOVB    (R1)+,(R0)      ; STORE NODE NAME
001226  122710  000040                  CMPB    #' ,(R0)        ; IS IT A SPACE?
001232  001403                          BEQ     20$             ; BR IF YES - END OF NODE NAME
001234  005200                          INC     R0              ; UPDATE R0
001236  005302                          DEC     R2              ; MORE TO MOVE?
001240  003371                          BGT     10$             ; BR IF YES
001242  112720  000042          20$:    MOVB    #'",(R0)+       ; STORE DOUBLE QUOTE AT END OF NAME
001246  105020                          CLRB    (R0)+           ; CREATE ASCIZ STRING
001250                                  RETURN

                                        .ENDC
```

```
                                        .SBTTL  STA$DF,ADD$DF STATE TABLES

                                ;
                                ; STA$DF
                                ;
                                        STATE$  STADF
                                         TRAN$   $NUMBR
                                        STATE$
                                         TRAN$   <',>
                                        STATE$
                                         TRAN$   $RAD50
                                        STATE$
                                         TRAN$   <',>,STADF1
                                         TRAN$   $LAMDA,$EXIT           ; DON'T REQUIRE ANYMORE
                                        STATE$  STADF1
                                         TRAN$   '!,STADF1
                                         TRAN$   %SF.ENA%,STADF1,,SF.ENA,MRKFL
                                         TRAN$   %SF.MFL%,STADF1,,SF.MFL,MRKFL
                                         TRAN$   %SF.SER%,STADF1,,SF.SER,MRKFL
                                         TRAN$   $RAD50,STADF1
                                         TRAN$   $LAMDA,$EXIT,X.FLG

                                ;
                                ; ADD$DF
                                ;
                                        STATE$  ADDDF
                                         TRAN$   $NUMBR,$EXIT
```

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES		
LR.PRM		000030		#5-69		
LR.PRO		000002		#5-69		
LR.STS		000000		#5-69		
LR.TCB		000002		#5-69		
LR.UNT		000005		#5-69		
LS.AST	=	100000		#5-69		
LS.CEX	=	000001		#5-69		
LS.CIR	=	000010		#5-69		
LS.CXO	=	010000		#5-69		
LS.CX5	=	000400		#5-69		
LS.DLM	=	004000		#5-69		
LS.ECH	=	001000		#5-69		
LS.FDX	=	004000		#5-69		
LS.HDX	=	010000		#5-69		
LS.LIN	=	000003		#5-69		
LS.LMC	=	000007		#5-69		
LS.NTI	=	004000		#5-69		
LS.ON	=	000011		#5-69		
LS.OPT	=	000400		#5-69		
LS.PRO	=	000002		#5-69		
LS.PWF	=	040000		#5-69		
LS.TOP	=	002000		#5-69		
LS.UNF	=	020000		#5-69		
LS.X25	=	000012		#5-69		
LS.11D	=	004000		#5-69		
LX.CEX	=	000004		#5-69		
LX.LIN	=	000006		#5-69		
LX.PRO	=	000005		#5-69		
L.COST		000015		#5-71		
L.CTL		000012		#5-71		
L.CVA		17777		#5-71		
L.DDM		000004		#5-71		
L.DDS		000005		#5-71		
L.DLC		000003		#5-71		
L.DLM		000006		#5-71		
L.DLS		000010		#5-71		
L.FLG		000000		#5-71	*8-322	
L.KRBA		000016		#5-71		
L.LEN	=	000022		#5-71		
L.MPF		000022		#5-71		
L.NMST		000020		#5-71		
L.NSTA		000014		#5-71		
L.OWNR		000021		#5-71		
L.UNT		000013		#5-71		
MFLDF		001026	R	11-498		
MRKFL		002454	RG	#7-250	*8-326	
MS.BUF	=	******	GX	8-392		
MS.NT1	=	******	GX	8-297	8-302	
MS.NT2	=	******	GX	8-297		
MS.PAR	=	******	GX	8-399		
M$$MGE	=	000000		7-120	7-131	22-928
NBRA		002450	RG	#7-240	8-338	

NTLCF2 CREATED BY MACRO ON 29-JUN-85 AT 00:27 PAGE 4

LOCAL DATA

```
    139 000374                          HSHSZ:: .BLKW   1               ; HASH TABLE SIZE
    140 000376                          HSHADD::.BLKW   1               ; ADDRESS OF HASH TABLE
    141 000400                          CTIM::  .BLKW   1               ; COUNTER TIMER VALUE
    142                                 ;
    143                                 ; DATA FOR PVC$DF MACRO
    144                                 ;
    145 000402                          PVSTA:: .BLKW   1               ; PVC STATE
    146         000001                          PV.OFF == 1             ;       STATE = OFF
    147         000002                          PV.ON  == 2             ;       STATE = ON
    148                                 ;
    149                                 ; DATA FOR DST$DF MACRO
    150                                 ;
    151 000404                          DSFLG:: .BLKW   1               ; LOCAL FLAG WORD
    152         000001                          DS.X25 == 1             ; X25 DESTINATION TYPE
    153         000002                          DS.X29 == 2             ; X29 DESTINATION TYPE
    154 000406                          DSTTSK::.BLKW   2               ; OBJECT TASK NAME - RAD50
    155 000412                          DSTNAM::.BLKB   6               ; DESTINATION NAME
    156 000420                          DSTPRI::.BLKB   1               ; PRIORITY
    157 000421                          DSTOBJ::.BLKB   1               ; OBJECT NUMBER
    158                                 ;
    159                                 ; TABLE OF ADDRESSES OF COUNT FIELDS FOR VARIABLE LENGTH PORTION
    160                                 ; OF DESTINATION DESCRIPTOR BLOCK
    161                                 ;       ENTRY FORMAT:
    162                                 ;               ADDRESS OF COUNT, ADDRESS OF SPECIFIED FIELD
    163                                 ;
    164                                 ;       *** ORDER IS IMPORTANT (SEE DSTDF$ MACRO) ***
    165                                 ;
    166 000422  000506' 000507'         DSTVAR::.WORD   DSCMCT,DSCMSK   ; CALL DATA MASK COUNT
    167 000426  000547' 000550'                 .WORD   DSCVCT,DSCVAL   ; CALL DATA VALUE COUNT
    168 000432  000500' 000460'                 .WORD   DSARCT,DSARDP   ; REMOTE DTE ADDRESS COUNT
    169 000436  000000                          .WORD   0               ; END OF TABLE
    170                                 ;
    171                                 ; DATA FOR DSA$DF
    172                                 ;
    173         000017                          DARMLN == 15.           ; MAX NUMBER OF DIGITS IN REMOTE DTE ADDRESS
    174 000440                          DSARDT::.BLKB   DARMLN          ; TEMP STORAGE FOR UNPACKED REMOTE DTE ADDRESS
    175 000457      000                         .BYTE   0               ; END OF STRING INDICATOR FOR REMOTE DTE ADDR
    176         000456'                         DARDEN == .-2           ; END OF REMOTE DTE ADDRESS
    177 000460                          DSARDP::.BLKB   <DARMLN/2>+1    ; PACKED DTE ADDRESS
    178 000470                          DSACUG::.BLKB   6               ; CLOSED USER GROUP NAME
    179 000476                          DSALCT::.BLKB   1               ; COUNT OF DIGITS IN LOW SUBADDRESS
    180 000477                          DSAHCT::.BLKB   1               ; COUNT OF DIGITS IN HIGH SUBADDRESS
    181 000500                          DSARCT::.BLKB   1               ; COUNT OF DIGITS IN REMOTE DTE ADDRESS
    182                                         .EVEN
    183 000502                          DSASLO::.BLKW   1               ; LOW END OF SUBADDRESS RANGE
    184 000504                          DSASHI::.BLKW   1               ; HIGH END OF SUBADDRESS RANGE
    185                                 ;
    186                                 ; DATA FOR DSC$DF MACRO
    187                                 ;
    188 000506                          DSCMCT::.BLKB   1               ; NUMBER OF HEX DIGITS IN CALL DATA MASK
    189 000507                          DSCMSK::.BLKB   32.             ; CALL DATA MASK
    190 000547                          DSCVCT::.BLKB   1               ; NUMBER OF HEX DIGITS IN CALL DATA VALUE
    191 000550                          DSCVAL::.BLKB   32.             ; CALL DATA VALUE
    192                                         .EVEN
```

```
                                        .SBTTL  STRNXT - STORE NEXT DIGIT

                                ;+
                                ;
                                ; STRNXT - STORE NEXT DIGIT
                                ;
                                ; INPUTS:
                                ;       .PCHAR - MATCHED CHARACTER FROM TPARS TRANSITION
                                ;       NEXT - ADDRESS OF NEXT BYTE AVAILABLE FOR STORAGE
                                ;
                                ; OUTPUTS:
                                ;       NEXT - ADDRESS OF NEXT AVAILABLE BYTE
                                ;       COUNT - UPDATED
                                ;       R0,R1 DESTROYED
                                ;-
001064  116767  000000G 000357' STRNXT::MOVB    .PCHAR,CVTBUF           ; MOVE DIGIT TO CONVERTSION BUFFER
001072  012701  000017          MOV     #17,R1                  ; ASSUME CHAR IS AN '*'
001076  126727  000000G 000052  CMPB    .PCHAR,#'*              ; IS CHAR AN '*' ?
001104  001404                  BEQ     10$                     ; BR IF YES
001106  012700  000357'         MOV     #CVTBUF,R0              ; POINT TO CONVERSION BUFFER
001112                          CALL    $CDTB                   ; CONVERT TO BINARY
001116  110177  000352'         10$:    MOVB    R1,@NEXT        ; STORE BINARY DIGIT
001122  005267  000352'                 INC     NEXT            ; POINT TO NEXT AVAILABLE BYTE
001126  105267  000356'                 INCB    COUNT           ; INCREMENT DIGIT COUNT
001132                          RETURN
                                ;
000001                          .END
```

```
                                        .ENDC
                                
000114  005426                  30$:    NEG     (SP)+           ; SET C-BIT IF DEVICE IS OFFLINE
000116                                  RETURN
                                
                                ;
                                ; MARK DEVICE OFFLINE
                                ;
000120  005266  000004          OFFLIN: INC     4(SP)           ; INDICATE FAILURE
000124  000002                          RTI                     ; RETURN FROM SST
                                
                                
        000001                          .END
```

```
                                ; ERROR CONDITION
                                ;
000110  012701  000146'  101$:  MOV     #DUMMY-4,R1     ; POINT AT DUMMY SYMBOL
000114                          EMSG$R  2Y              ; DCB DISPATCH TABLE ADDRESS UNDEFINED
                                
                                .IFF    ; DF    R$$11M
                                
                                CLR     R0              ;
                                MOV     $$DEV,R4        ; GET ASCII DEVICE NAME
                                CLR     R5              ; UNIT NUMBER 0
                                CALL    ..FDEV          ; LOOK FOR DEVICE
                                BCS     102$            ; BR IF CAN'T FIND IT
                                MOV     #DEVNAM,R0
                                MOVB    $$DEV,(R0)      ; GET FIRST CHARACTER OF DEVICE NAME
                                MOVB    $$DEV+1,1(R0)   ; GET SECOND CHARACTER
                                MOV     PC,R1           ; PERIODS ARE VALID
                                CALL    $CAT5           ; CONVERT IT TO RAD50
                                MOV     #REQUES,R0      ; POINT TO RQST$ DPB
                                MOV     R1,R.QSTN(R0)   ; STORE FIRST HALF OF TASK NAME
                                DIR$    R0              ; REQUEST TASK TO RUN
                                BCS     101$            ; BR IF FAILED
                                MOV     $FEATR,U.C3(R5) ; STORE FEATURES WORD
                                MOV     $GCD,U.LBH(R5)  ; STORE GCD ADDRESS OF PROCESS
                                RETURN
                        101$:   EMSG$R  2A              ; REQUEST DRIVER TO RUN
                        102$:   EMSG$R  2B              ; DEVICE NOT IN SYSTEM
```

```
                                        .SBTTL  REP9 - REPLACE DEVICE NAME
                                ;+
                                ; REP9 - REPLACE DEVICE NAME
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
001600  112314                  REP9:   MOVB    (R3)+,(R4)      ; COPY THE FORMAT STRING
001602  122724  000052                  CMPB    #'*,(R4)+       ; LOOKING FOR THE FIRST ASTERISK
001606  001374                          BNE     REP9            ;       ...
001610  116764  000000G 177777          MOVB    $$DEV,-1(R4)    ; STORE DEVICE NAME
001616  116724  000001G                 MOVB    $$DEV+1,(R4)+   ;       ...
001622                                  RETURN
```

MAPx CREATED BY MACRO ON 30-JUN-85 AT 00:00 AGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES	
CALL	5-93	
RESRG	#4-45	6-145
RETURN	5-95	6-146
SAVRG	#4-45	6-113

FILEID**NTL

```
                                        .SBTTL  MACRO DEFINITIONS

                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  ASL$,ASR$,CCBDF$,OPTDF$,EMSG$R,DHBDF$,CTRDF$,RETC,BIAS5
                                        .MCALL  XPDDB$,MAP,SLTDF$,RESRG,SAVRG

000000                                  CTRDF$                  ; DEFINE LINE COUNTER BLOCK OFFSETS
000000                                  SLTDF$                  ; DEFINE SLT OFFSETS
000000                                  XPDDB$                  ; DEFINE XPT DATA BASE OFFSETS

                                        .MCALL  NHWDF$

000000                                  NHWDF$                  ; DEFINE HARDWARE OFFSETS

                                ;
                                ; LIBRARY SYMBOLS
                                ;
000000                                  CCBDF$                  ; CCB OFFSETS
000000                                  OPTDF$                  ; COMM/EXEC OPTIONS
000000                                  DHBDF$                  ; DECNET HOME BLOCK OFFSETS

                                ;
                                ; LOCAL MACRO DEFINITIONS
                                ;
                                        .MACRO  SAVMAP
                                        MOV     @KSAR5,-(SP)    ; SAVE APR 5
                                        .ENDM

                                        .MACRO  RESMAP
                                        MOV     (SP)+,@KSAR5    ; RESTORE APR5
                                        .ENDM

                                        .MACRO  MAP BIAS
                                        MOV     BIAS,@KSAR5
                                        .ENDM

                                        .MACRO  BIAS5 REG
                                        BIC     #160000,REG
                                        BIS     #120000,REG
                                        .ENDM
```

```
                                        .SBTTL  CECLK - SET UP MULTI-PROCESSOR CLOCK QUEUE ENTRIES

                                ;+
                                ;
                                ; CECLK - SET UP MULTI-PROCESSOR CLOCK QUEUE ENTRIES
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
001354                          CECLK:  .IF DF  R$$MPL

                                        MOVB    @NCPU,R0        ; GET # OF PROCESSORS IN CONFIGURATION
                                        BEQ     100$            ; IF EQ, SINGLE PROCESSOR SYSTEM

                                        SWSTK$  100$            ; ENTER SYSTEM STATE
                                        CLR     R1              ;; INITIALISE PROCESSOR NUMBER

                                10$:    MOV     K6TAB,R2        ;; GET $K6TAB ADDRESS
                                        ADD     R1,R2           ;; POINT TO CPU PARTITION BIAS
                                        MOV     (R2),@KSAR5     ;; MAP TO THE CPU PARTITION
                                        MOV     M100Q,R3        ;; GET $M100Q ADDRESS
                                        MOV     R3,R4           ;; SAVE ASSRESS
                                        ADD     #120000,R3      ;; MAP THRU APR5
                                        CLR     (R3)+           ;; INITIALISE 100 MSEC TIMER QUEUE
                                        MOV     R4,(R3)         ;; ...
                                        ADD     #2,R1           ;; MOVE ON PROCESSOR NUMBER
                                        SOB     R0,10$          ;; SCAN ALL PROCESSORS

                                100$:   RETURN

                                        .ENDC
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ALOCB	= ****** GX	9-506
BUFUMP	= 172354	#5-68
CBLK1	000040 R	#6-175 8-405 8-464
CBLK3	000054 R	6-175 #6-179
CBL.4	000070 R	6-179 #6-184
CBLK5	000104 R	6-184 #6-189
CB.CCB	= 000002	14-843
CCBCT	= ****** GX	13-723 13-726
CCBLH	= ****** GX	14-841 14-842 14-851 14-853
CCBNM	= ****** GX	13-719 13-727 14-823
CCBS	001736 R	13-788 #14-822
CCBSZ	= ****** GX	14-824 14-854
CEAVL	= ****** GX	7-326
CECLK	001354 R	#12-678
CENTI	001354 R	11-660 #13-718
CEPTR	001012 R	7-356 #11-596
CEPWR	= ****** GX	7-365
CESYM	000324 R	7-303 #8-398
CETNAM	000012 R	#6-143 7-347 8-421
CEXNAM	000006 R	#6-131 7-308 8-400 8-414
CEXPCB	000036 R	#6-170 *7-325 10-559
CF.DDM	= 000002	#5-74
CF.DYN	= 000004	#5-74 13-724
CF.EIS	= 000010	#5-74
CF.FRK	= 100000	#5-74
CF.LOG	= 000020	#5-74
CF.MDM	= 000001	#5-74
CF.TIM	= 000400	#5-74
CMODE	= 140000	#5-68
CRC$OV	= ****** GX	15-886 16-933
CXOPT	= ****** GX	11-608 13-724
C.BID	000003	*14-843
DECBLK	000552 R	7-354 #9-501
DECPT	= ****** GX	7-367 9-512 9-513 11-610
D$AMXC	000072	*11-618
D$AMXH	000074	*11-619
D$BRTM	000100	*11-620
D$END	= 000104	9-503
D$HIOR	000024	*9-517 *9-518 *9-519 *9-520
D$LNAM	000006	7-368
D$MAXC	000064	*11-613
D$MAXH	000066	*11-614
D$MAXV	000070	*11-615 *11-621
D$NA	000062	*11-616
D$NBEA	000056	*11-617
D$NN	000060	*11-611
D$RNN	000002	9-515 *9-516
D$RTMR	000076	*11-612
ELTIME	= ****** GX	7-358
ETAB1	000334 R	#6-256 8-483
ETAB2	000344 R	#6-264 7-377
E$NBR	000014	#5-62

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 2

```
                                        .SBTTL LOCAL DATA

                                ;
                                ; LOCAL DATA
                                ;
000000                                  .PSECT  DATA,D

                                        .NLIST  BEX

                                ;
                                ; CONFIG FILE NAME AND EXTENSION
                                ;
000000  103  105  124  CFGNAM: .ASCIZ  "CETAB"
000006  115  101  103  CFGEXT: .ASCIZ  "MAC"


                                ;
                                ; LINE IDENTIFICATION
                                ;
000012                          NAME::  .BLKB   3               ; DDM NAME
000015                          CTL::   .BLKB   1               ; CONTROLLER NUMBER
000016                          UNT::   .BLKB   1               ; UNIT NUMBER

                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
000017  015                     ERRMSG::.BYTE   15
000020  116  124  114                  .ASCII  "NTL -- "
000027                          ERRBUF::.BLKB   70.

                                        .EVEN

000136                          CFGFLG::.BLKW   1               ; FLAGS WORD
                000001                  FG.X2P == 1             ; X2P$DF MACRO IS NECESSARY
                000002                  FG.X3P == 2             ; X3P$DF MACRO IS NECESSARY

                                ;
                                ; COUNTERS
                                ;
000140                          SCNT::  .BLKW   1               ; STA$DF'S SEEN
000142                          UCNT::  .BLKW   1               ; UNT$DF'S SEEN


                                ;
                                ; RECORD BUFFER AND BUFFER LENGTH
                                ;
000144                          CFGBF:: .BLKB   134.
000352                          CFGSZ:: .BLKW   1

                                ;
                                ; SAVED STACK POINTER FOR ERROR EXIT
                                ;
000354                          SPSAV:  .BLKW   1

                                ;
                                ; UNIBUS RUN MASK FLAG
                                ;
                                        .IF DF  R$$MPL
                                URMFL:: .BLKW   1               ; =1 IF UNIBUS RUN MASK PRESENT
                                        .ENDC
```

```
                                        .SBTTL  TPARS STATE TABLES - COMMON SUBEXPRESSIONS
                                ;
                                ; TPARS STATE TABLES
                                ;
001402                                  ISTAT$  CFGST,CFGKW
                                ;
                                ; COMMON SUBEXPRESSIONS
                                ;
                                ;
                                ; FLAG BIT DEFINITIONS PLUS TRAILING COMMA
                                ;
001402                                  STATE$  BITS
001402                                   TRAN$   %LF.X2P%,BITS1,,FG.X2P,CFGFLG
001402                                   TRAN$   %ZF.X3P%,BITS1,,FG.X3P,CFGFLG
001402                                   TRAN$   $RAD50,BITS1
001402                                   TRAN$   <',>,$EXIT
001402                                  STATE$  BITS1
001402                                   TRAN$   <'!>,BITS2
001402                                   TRAN$   '+,BITS2
001402                                   TRAN$   <',>,$EXIT
001402                                  STATE$  BITS2
001402                                   TRAN$   %LF.X2P%,BITS1,,FG.X2P,CFGFLG
001402                                   TRAN$   %ZF.X3P%,BITS1,,FG.X3P,CFGFLG
001402                                   TRAN$   $RAD50,BITS1
                                ;
                                ; CHECK FOR END OF SOURCE LINE
                                ;
001402                                  STATE$  END
001402                                   TRAN$   <';>,$EXIT
001402                                   TRAN$   $EOS,$EXIT
```

```
                                                                M 7
NTLCFG - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:23  Page 27-2
Symbol table

$NAME = ****** GX       $QPVC = ****** GX       $REP.N= ****** GX       $$INC = ****** GX       $$$TMP= 000052R     005
$NLCFG  000000RG        $QSLT = ****** GX       $SLTA = ****** GX       $$LSAD= ****** GX       .PCHAR= ****** GX
$NUMBR= 000002          $QSTA = ****** GX       $STRNG= 000004          $$MTP = ****** GX       .PNUMB= ****** GX
$PDVA = ****** GX       $QSVC = ****** GX       $SUBXP= 000010          $$MUX = ****** GX       .PNUMH= ****** GX
$QCNT = ****** GX       $QUNT = ****** GX       $TIOUT= ****** GX       $$$FLG= 177777          .PSTCN= ****** GX
$QDDM = ****** GX       $QX2P = ****** GX       $$DEV = ****** GX       $$$KEY= 000006          .PSTPT= ****** GX
$QDTE = ****** GX       $QX3P = ****** GX       $$EXT = ****** GX       $$$STA= 000000          .TPARS= ****** GX
$QFEA = ****** GX       $RAD50= 000016

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        002140    001   (RW,I,LCL,REL,CON)
DATA    001730    002   (RW,D,LCL,REL,CON)
$STATE  000260    003   (RW,D,LCL,REL,CON)
$KTAB   000016    004   (RW,D,LCL,REL,CON)
$KSTR   000061    005   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 15122 Words  ( 60 Pages)
Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:13.21
SY:NTLCFG.V2,[132,134]NTLCFG/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLCFG
```

```
                                                                N 7
NTLCFG     CREATED BY  MACRO  ON 29-JUN-85 AT 00:24     PAGE 1
```

```
                                        ;
                                                .ENABL  LC
                                                .NLIST  BEX
001710      052     127     101 FMT9::  .ASCIZ  "*WARNING* -- "
001726      103     157     156 FMT8::  .ASCIZ  "Config File -- "
001746      103     157     156 FMT8B:: .ASCIZ  "Config File -- Partition * "
                                                .IF NDF R$$11M
                                        FMT8C:  .ASCIZ  "Config File -- Region *, "
                                                .ENDC

                                                .EVEN
                                                .DSABL  LC
                                                .LIST   BEX


002002                                  STAN::  .BLKW   1               ; STATION COUNT
002004  000000                          $LSTRM::.WORD   0               ; LAST REMOTE NODE BLOCK POINTER
002006  000000                          $LSTAD::.WORD   0               ; HIGHEST NODE ADDRESS

                                        ;+
                                        ; DLL DATA
                                        ;-

002010                                  SERBEG::
002010                                  SERFLG::.BLKW   1               ; SER$DF PARSING FLAG
002012                                  SERNOD::.BLKW   1               ; NODE ADDRESS
002014                                  SERHST::.BLKW   1               ; HOST ADDRESS
002016                                  SERCIR::.BLKW   1               ; SLN/STA PAIR
002020                                  SERDEV::.BLKW   1               ; DEVICE NAME (RAD50)

002022                                  SERPAS::.BLKB   8.              ; PASSWORD
002032                                  ENDPAS::
002032                                  SERDMP::.BLKW   2               ; DUMP ADDRESS (22 BITS)
002036                                  SERDCT::.BLKW   1               ; DUMP COUNT
002040                                  SERADD::.BLKW   3               ; HARDWARE ADDRESS
002046                                  SERTMP::.BLKW   1               ; WORK LOCATION

002050                                  CIRNAM::.BLKW   2               ; CIRCUIT NAME
002054                                  CIRNUM::.BLKW   1               ; CIRCUIT NUMBER
002056                                  CIRMUL::.BLKW   1               ; TRIB NUMBER
002060                                  DEVNM:: .BLKW   2               ; DEVICE NAME (ASCII)
002064                                  HEXSTR::.BLKB   16.             ; HEX STRING BUFFER
002104                                  ENDHEX::

002104                                  STRSIZ::.BLKW   1               ; STRING LENGTH
002106                                  UICBUF::.BLKW   6               ; UIC BUFFER
        000014                          UISZ==.-UICBUF                  ; UIC BUFFER SIZE
002122                                  FILBUF::.BLKB   16.             ; FILE NAME BUFFER
        000020                          FLSZ==.-FILBUF                  ; FILE NAME BUFFER SIZE

002142                                  DEVBUF::.BLKW   4               ; DEVICE NAME BUFFER
        000010                          DVSZ==.-DEVBUF                  ; DEVICE NAME BUFFER SIZE
002152                                  FILFLG::.BLKW   1               ; FIL$DF PARSING FLAG
002154                                  CIRUNT::.BLKW   1               ; UNIT NUMBER OF CIRCUIT
002156                                  SERCUR::.BLKW   1               ; CURRENT SERVICE BLOCK UNMAPPED ADDRESS
002160                                  SEROFF::.BLKW   1               ; FNB POINTER OFFSET INTO SERVICE BLOCK
```

```
694
695
696
697                                     ;+
698                                     ; FIL$DF STATE TABLE
699                                     ;-
700
701 001252                                      ISTAT$  CFGST,CFGKW
702
703
704 001252                                      STATE$  FILDF
705 001252                                       TRAN$  <',>,TER
706 001252                                       TRAN$  $EOS,$EXIT
707 001252                                       TRAN$  !FILSPC,,.FLBK0
708
709 001252                                      STATE$
710 001252                                       TRAN$  $EOS,$EXIT
711 001252                                       TRAN$  <',>
712
713 001252                                      STATE$ TER
714 001252                                       TRAN$ <',>,LOA
715 001252                                       TRAN$  !FILSPC,,.FLBK1
716
717 001252                                      STATE$
718 001252                                       TRAN$  $EOS,$EXIT
719 001252                                       TRAN$  <',>
720
721 001252                                      STATE$  LOA
722 001252                                       TRAN$  <',>,DIA
723 001252                                       TRAN$  !FILSPC,,.FLBK2
724
725 001252                                      STATE$
726 001252                                       TRAN$  $EOS,$EXIT
727 001252                                       TRAN$  <',>
728
729 001252                                      STATE$  DIA
730 001252                                       TRAN$  <',>,DUM
731 001252                                       TRAN$  !FILSPC,,.FLBK3
732
733 001252                                      STATE$
734 001252                                       TRAN$  $EOS,$EXIT
735 001252                                       TRAN$  <',>
736
737 001252                                      STATE$ DUM
738 001252                                       TRAN$  !FILSPC,,.FLBK4
739
740 001252                                      STATE$
741 001252                                       TRAN$  $EOS,$EXIT
742
```

```
                                        .SBTTL  TPARS STATE TABLES - OBJ$DF
                                ;
                                ; OBJECT DEFINITIONS (OBJ$DF)
                                ;
001252                          STATE$  OBJDF
001252                           TRAN$   $NUMBR,,O.NUM

001252                          STATE$
001252                           TRAN$   <',>

001252                          STATE$
001252                           TRAN$   ',>,OBJ2                        ; ALLOW FOR BLANK OBJECT NAME
001252                           TRAN$   LBRA

001252                          STATE$
001252                           TRAN$   $RAD50,,O.NAME

001252                          STATE$
001252                           TRAN$   RBRA

001252                          STATE$
001252                           TRAN$   <',>

001252                          STATE$  OBJ2
001252                           TRAN$   $NUMBR,,O.VFY

001252                          STATE$
001252                           TRAN$   <',>

001252                          STATE$
001252                           TRAN$   $NUMBR,,O.FLAG

001252                          STATE$
001252                           TRAN$   !END,$EXIT,O.END
001252                           TRAN$   <',>

001252                          STATE$
001252                           TRAN$   $NUMBR,$EXIT,O.CPY
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
OPEN	000510 R	8-300 #9-431
PARDF	000654 R	11-495
PDVNM	= ****** GX	8-310
PDVTA	= ****** GX	8-308
QBUF	000644 R	8-390 #11-493
QMFL	000660 R	8-327 #11-498
QPAR	000652 R	8-383 8-397 #11-495
QROU	= ****** GX	8-388 8-395 8-402
RBRA	= 000076	#7-264
READ	000552 R	8-324 8-381 8-393 8-400 #10-459
REPND	001174 RG	7-114 #16-674
REP.NN	001136 RG	#15-645
REP&B	001066 RG	7-112 7-113 7-117 7-119 7-121 7-133 #14-613 14-616
REP&C	001066 RG	#14-614
RTSPC	001044 RG	7-109 7-110 7-122 7-127 7-128 7-129 7-132 7-134 7-135
		7-141 7-142 7-143 7-144 7-145 7-146 7-147 7-148 7-149
		7-150 7-151 12-568 #12-571
R$$11M	= 000000	5-62 7-111 7-164 8-331 15-673
SERADD	002040 RG	#7-192
SERBEG	002010 RG	#7-181
SERCIR	002016 RG	#7-185
SERCUR	002156 RG	#7-212
SERDCT	002036 RG	#7-191
SERDEV	002020 RG	#7-186
SERDMP	002032 RG	#7-190
SERFIN	002166 RG	#7-217
SERFLG	002010 RG	#7-182
SERHST	002014 RG	#7-184
SERNOD	002012 RG	#7-183
SEROFF	002160 RG	#7-213
SERPAS	002022 RG	#7-188
SERSLT	002162 RG	#7-215
SERTMP	002046 RG	#7-193
SF.ACT	= 000200	#5-71
SF.CIR	= 000010	#5-73
SF.DEV	= 000020	#5-73
SF.DIA	= 004000	#5-73
SF.DPA	= 000100	#5-73
SF.DPC	= 000200	#5-73
SF.DUM	= 010000	#5-73
SF.ENA	= 000100	#5-71
SF.HAD	= 000004	#5-73
SF.HST	= 000002	#5-73
SF.LOA	= 002000	#5-73
SF.LPB	= 000004	#5-71
SF.MFL	= 000040	#5-71
SF.PAC	= 000020	#5-71
SF.PH3	= 020000	#5-73
SF.PSS	= 000040	#5-73
SF.REA	= 000010	#5-71
SF.SEC	= 000400	#5-73
SF.SER	= 000001	#5-71

```
                                        .SBTTL  ERROR MESSAGES
                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000610                                  NTLER$  <:>,P7,10,CERR,RTSPC,CFLIN,<default block size>
000644                                  NTLER$  <:>,P8,10,CERR,RTSPC,CFLIN,<maximum block size>
000700                                  NTLER$  <:>,P9,10,CERR,RTSPC,CFLIN,<default window size>
000734                                  NTLER$  <:>,00,10,CERR,RTSPC,CFLIN,<maximum window size>
000770                                  NTLER$  <:>,01,8,CERR,RTSPC,CFLIN,<Default block size exceeds maximum>
001044                                  NTLER$  <:>,02,8,CERR,RTSPC,CFLIN,<Default window size exceeds maximum>
001120                                  NTLER$  <:>,05,10,CERR,RTSPC,CFLIN,<counter timer value>
001154                                  NTLER$  <:>,06,10,CERR,RTSPC,CFLIN,<line-id>
001174                                  NTLER$  <:>,09,10,CERR,RTSPC,CFLIN,<logical channel number>
001234                                  NTLER$  <:>,M4,8,CERR,RTSPC,,<X3P$DF missing>
001264                                  NTLER$  <:>,M5,8,CERR,RTSPC,,<DSA$DF missing>
001314                                  NTLER$  <:>,M6,8,CERR,RTSPC,,<DSC$DF missing>
001344                                  NTLER$  <:>,1P,8,CFERR,$REP.N,$FNOUT,<Open Failure (-***.)>
001402                                  NTLER$  <:>,1Q,8,CFERR,$REP.N,$FNOUT,<Read Failure (-***.)>
001440                                  NTLER$  <:>,1R,8,CFERR,RTSPC,FNLIN,<Read Overflow>
001466                                  NTLER$  <:>,1T,8,CERR,RTSPC,CFLIN,<Syntax Error>
                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
001514  103     157     156  FMT8::  .ASCIZ  "Config File -- "
001534  103     157     156  FMT10:: .ASCIZ  "Config File -- Illegal "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

000000                                  .PSECT
```

Symbol table

```
A$$CHK= 000000          D$$ISK= 000000          LN.ODP= 000004          L.DDS   000004          STRNXT  001064RG
A$$CPS= 000000          D$$L11= 000001          LN.DPE= 000001          L.DLC   000003          SUBAMN= 000000 G
A$$PRI= 000000          D$$YNC= 000000          LN.REF= 000002          L.DLM   000006          SUBAMX= 023417 G
A$$TRP= 000000          D$$YNM= 000000          LN.SER= 000002          L.DLS   000010          SYNERR  000342RG    002
BLKSMN= 000040          ERRBUF  000022R   002   LN.STA= 000017          L.FLG   000000          S$$WRG= 000000
BLKSMX= 002000          ERRMSG  000012R   002   LN.SUB= 000360          L.KABA  000016          S$$YSZ= 007600
BLKSZD  000360RG  002   E$$XPR= 000000          LN.TRI= 000006          L.LEN = 000022          S.COST  000001
BLKSZM  000362RG  002   FL.NTI= ****** GX       LO.CON  000006          L.MPF   000022          S.FLG   000000
CERR    000374RG        FMT10   001534RG  002   LO.CSR  000006          L.NMST  000020          S.LEN   000004
CFERR   000404RG        FMT8    001514RG  002   LO.CTL  000014          L.NSTA  000014          S.NMST  000002
CFGBF   000130RG  002   FM.10 = 000000          LO.LIN  000004          L.OWNR  000021          S.OWNR  000003
CFGEXT  000006R   002   FM.8  = 000000          LO.PAR  000010          L.UNT   000013          T$$KMG= 000000
CFGNAM  000000R   002   FNDPDV  001012RG        LO.PRI  000010          MRKFL   000344RG  002   T$$MIN= 000000
CFGSZ   000336RG  002   FNLIN   000500RG        LO.SCR  000014          M$$CRB= 000124          V$$CTR= 001000
CFLIN   000504RG        F$$LVL= 000001          LO.STA  000015          M$$CRX= 000000          WNDSMX= 000177
CHKBLK  000610RG        G$$TPP= 000000          LO.VCT  000012          M$$FCS= 000000          WNDSZD  000364RG    002
CHKWND  000646RG        G$$TSS= 000000          LR.AST  000026          M$$MGE= 000000          WNDSZM  000366RG    002
CHNLMX= 007777 G        G$$TTK= 000000          LR.CIR  000002          M$$NET= 000000          X$$DBT= 000000
COUNT   000356RG  002   G$$WRD= 000000          LR.CTL  000004          M$$DVR= 000000          Z.NAM = ****** GX
CTIM    000400    002   HSHADD  000376RG  002   LR.EFN  000004          NEXT    000352RG  002   $AZFS = ****** GX
CUNMMX= 000006          HSHSZ   000374RG  002   LR.LIN  000002          N$$ACC= 000001          $CAT5 = ****** GX
CVTBUF  000357RG  002   IE.EDF= ****** GX       LR.PRM  000030          N$$BUF= 000001          $CDTB = ****** GX
CWARN   000364R         IE.RBG= ****** GX       LR.PRO  000002          N$$LDV= 000001          $CEACX= ****** GX
C$$CKP= 000000          I$$RAR= 000000          LR.STS  000000          N$$MCP= 000001          $CFUIC= ****** GX
C$$ORE= 000400          I$$RDN= 000000          LR.TCB  000002          N$$MLL= 000001          $CLBUF= ****** GX
C$$RSH= 177564          K$$CNT= 177546          LR.UNT  000005          N$$MOV= 000010          $CLDEQ= ****** GX
DARDEN= 000456RG  002   K$$CSR= 177546          LS.AST= 100000          N$$NCT= 000001          $CLIDS= ****** GX
DARMLN= 000017 G        K$$LDC= 000000          LS.CEX= 000001          N$$PEM= 000001          $CLQUE= ****** GX
DEVCTL  000546RG        K$$TPS= 000074          LS.CIR= 000010          OPEN    000230R         $CLSIZ= ****** GX
DEVNAM  000520RG        LBRA  = 000074          LS.CXO= 010000          PCKBCD  000700RG        $ERRM4  001234RG    002
DEVUNT  000564RG        LD$LP = 000000          LS.CX5= 000400          PDVNM = ****** GX       $ERRM5  001264RG    002
DE.OFF= 000002 G        LF.ACT= 100000          LS.DLM= 004000          PDVTA = ****** GX       $ERRM6  001314RG    002
DE.ON = 000001 G        LF.BRO= 000400          LS.ECH= 001000          PDRTND  000354RG  002   $ERRD0  000734RG    002
DSACUG  000470RG  002   LF.BWT= 000007          LS.FDX= 004000          PVSTA   000402RG  002   $ERRD1  000770RG    002
DSAHCT  000477RG  002   LF.ENA= 002000          LS.HDX= 010000          PV.OFF= 000001 G        $ERRD2  001044RG    002
DSALCT  000476RG  002   LF.LPB= 001000          LS.LIN= 000003          PV.ON = 000002 G        $ERRD5  001120RG    002
DSARCT  000500RG  002   LF.MDC= 000100          LS.LMC= 000007          P$$P45= 000000          $ERRD6  001154RG    002
DSARDP  000460RG  002   LF.MFL= 004000          LS.NTI= 004000          P$$WRD= 000000          $ERRD9  001174RG    002
DSARDT  000440RG  002   LF.MTP= 000020          LS.ON = 000011          Q$$OPT= 000010          $ERRP7  000610RG    002
DSASHI  000504RG  002   LF.PAC= 000200          LS.OPT= 000400          RBRA  = 000076          $ERRP8  000644RG    002
DSASLO  000502RG  002   LF.RDY= 040000          LS.PRO= 000002          READ    000272R         $ERRP9  000700RG    002
DSCMCT  000506RG  002   LF.REA= 010000          LS.PWF= 040000          RTSPC   000476RG        $ERR1P  001344RG    002
DSCMSK  000507RG  002   LF.SER= 000040          LS.TDP= 002000          R$$DER= 000000          $ERR1Q  001402RG    002
DSCVAL  000550RG  002   LF.TIM= 000010          LS.UNF= 020000          R$$K11= 000001          $ERR1R  001440RG    002
DSCVCT  000547RG  002   LF.UNL= 020000          LS.X25= 000012          R$$SND= 000000          $ERR1T  001466RG    002
DSFLG   000404RG  002   LF.X2P= 000000          LS.11D= 004000          R$$11M= 000000          $FLAGS= ****** GX
DSTEND= ****** GX       LINCTL  000350RG  002   LX.CEX= 000004          SF.ACT= 000200          $FNDUT= ****** GX
DSTNAM  000412RG  002   LINNAM  000346RG  002   LX.LIN= 000006          SF.ENA= 000100          $GCLEP= ****** GX
DSTOBJ  000421RG  002   LINUNT  000351RG  002   LX.PRO= 000005          SF.LPB= 000004          $LINKX= ****** GX
DSTPRI  000420RG  002   LNKEND  000760RG        L$$ASG= 000000          SF.MFL= 000040          $NTCF3  000000RG
DSTTSK  000406RG  002   LN.CLO= 000000          L$$DRV= 000000          SF.PAC= 000020          $QCHN = ****** GX
DSTVAR  000422RG  002   LN.DUM= 000005          L$$P11= 000001          SF.REA= 000010          $QCUG = ****** GX
DS.X25= 000001 G        LN.LOA= 000004          L$$11R= 000000          SF.SER= 000001          $QDSA = ****** GX
DS.X29= 000002 G        LN.LOD= 000003          L.COST  000015          SF.SVC= 000002          $QDSC = ****** GX
DTEDES  000370RG  002   LN.DAU= 000003          L.CTL   000012          SF.UNL= 000040          $QDST = ****** GX
DTEFLG  000372RG  002   LN.OFF= 000001          L.CVA   177776          SPACE = 000040          $QDTE = ****** GX
D$$BUG= 177514          LN.ON = 000000          L.DDM   000002          SPSAV   000340R   002   $QPSN = ****** GX
```

NTLCSR - CHECK CSR ADDRESS MACRO V05.03b Saturday 29-Jun-85 00:30 Page 6-2
Symbol table

```
A$$CHK= 000000          G$$TPP= 000000          L$$P11= 000001          N$$MOV= 000010          S$$WRG= 000000
A$$CPS= 000000          G$$TSS= 000000          L$$11R= 000000          N$$NCT= 000001          S$$YSZ= 007600
A$$PRI= 000000          G$$TTK= 000000          M$$CRB= 000124          N$$PEM= 000001          S.VTTA= 000002
A$$TRP= 000000          G$$WRD= 000000          M$$CRX= 000000          OFFLIN  000120R         S.VTTL= 000004
C$$CKP= 000000          I$$RAR= 000000          M$$FCS= 000000          PR7     = ****** GX     T$$KMG= 000000
C$$ORE= 000400          I$$RDN= 000000          M$$MGE= 000000          P$$P45= 000000          T$$MIN= 000000
C$$RSH= 177564          KSAR7 = ****** GX       M$$NET= 000000          P$$WRD= 000000          UAR7SV  000014R     002
D$$BUG= 177514          K$$CNT= 177546          M$$OVR= 000000          Q$$OPT= 000010          UDR7SV  000016R     002
D$$ISK= 000000          K$$CSR= 177546          NOSST   000006R     002 R$$DER= 000000          USAR7 = ****** GX
D$$L11= 000001          K$$LDC= 000000          N$$ACC= 000001          R$$K11= 000001          USDR7 = ****** GX
D$$YNC= 000000          K$$TPS= 000074          N$$BUF= 000001          R$$SND= 000000          V$$CTR= 001000
D$$YNM= 000000          LD$LP = 000000          N$$LDV= 000001          R$$11M= 000000          X$$DBT= 000000
E$$XPR= 000000          L$$ASG= 000000          N$$MCP= 000001          SST     000000R     002 $NLCSR  000000RG
F$$LVL= 000001          L$$DRV= 000000          N$$MLL= 000001          SSTAB   000020R     002 $$$OST= 000006

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000126    001   (RW,I,LCL,REL,CON)
DATA    000024    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics

Work  file  reads: 0
work  file writes: 0
Size of work file: 9248 Words  ( 37 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.28
SY:NTLCSR.V2,[132,134]NTLCSR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLCSR
```


NTLCSR CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 1

```
                                        .SBTTL  FIND - DETERMINE IF DATA STRUCTURES EXIST

                                ;+
                                ; FIND - DETERMINE IF DATA STRUCTURES ALREADY EXIST
                                ;
                                ; THIS ROUTINE IS CALLED TO LOOK FOR AN EXISTING DEVICE WHOSE NAME MATCHES
                                ; $$DEV. IF SUCH A DEVICE IS LOCATED, ITS DCB ADDRESS IS STORED IN FOUND.
                                ;
                                ; INPUTS:
                                ;       $$DEV=2 CHARACTER DEVICE NAME
                                ;
                                ; OUTPUTS:
                                ;       FOUND=DCB ADDRESS (IF SUCCESSFUL)
                                ;-
                                        .IFT    ; DF R$$11M

000122  005000                  FIND:   CLR     R0              ; INDICATE NO DCB (YET)
000124                                  SWSTK$  30$             ;* ENTER KERNEL MODE
000130  016700  000000G                 MOV     DEVHD,R0        ;* GET DEVICE POINTER
000134  011000                  10$:    MOV     (R0),R0         ;* GET NEXT DEVICE IN LIST
000136  001406                          BEQ     20$             ;* IF EQ, END OF LIST
000140  026760  000000G 000000G         CMP     $$DEV,D.NAM(R0) ;* DOES DEVICE NAME MATCH?
000146  001372                          BNE     10$             ;* IF NE, NO
000150  010066  000002                  MOV     R0,2(SP)        ;* SET USER MODE R0
000154                          20$:    RETURN                  ;* BACK TO USER MODE
000156  010067  000166'         30$:    MOV     R0,FOUND        ; IS THERE ALREADY SUCH A DEVICE?
000162  001416                          BEQ     40$             ; IF EQ, NO
000164  005760  000000G                 TST     D.DSP(R0)       ; IS THE DRIVER RESIDENT?
000170  001014                          BNE     101$            ; IF NE, YES - VERY BAD
000172  005001                          CLR     R1              ; GET FIRST UNIT NUMBER
000174  156001  000000G                 BISB    D.UNIT(R0),R1   ;
000200  005002                          CLR     R2              ; AND LAST UNIT NUMBER
000202  156002  000001G                 BISB    D.UNIT+1(R0),R2 ;
000206  160102                          SUB     R1,R2           ; CALCULATE UCB COUNT
000210  005202                          INC     R2              ;
000212  003406                          BLE     111$            ; IF LE, D.UNIT IS SCREWED UP
000214  010267  000172'                 MOV     R2,UNITS        ; SAVE UNIT COUNT
000220                          40$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000222                          101$:   EMSG$R  2X              ; DEVICE DRIVER ALREADY RESIDENT!
000230                          111$:   EMSG$R  2Q              ; ILLEGAL UNIT (UCB) COUNT
```

```
                                        .SBTTL  REPDV
                                ;+
                                ; REPDV - REPLACE DEVICE NAME
                                ;
                                ; INPUTS:
                                ;       R5 - ERROR MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       ** IN ERROR MESSAGE BLOCK REPLACED BY DEVICE NAME
                                ;       R0 DESTROYED
                                ;
                                ;-
                                        .IFF    ; DF R$$11M

                                REPDV:  MOV     R5,R0           ; POINT TO ERROR MESSAGE BLOCK ADDRESS
                                        TST     (R0)+           ; ...
                                10$:    CMPB    #'*,(R0)+       ; FOUND ASTERISK?
                                        BNE     10$             ; BR IF NO
                                        DEC     R0              ; BACK UP OVER ASTERISK
                                        MOVB    $$DEV,(R0)+     ; STORE FIRST HALF OF NAME
                                        MOVB    $$DEV+1,(R0)    ; STORE SECOND HALF OF NAME
                                        RETURN
                                        .ENDC   ; DF    R$$11M

000001                                  .END
```

```
**FILE**ID**NTL
```

```
NN       NN   TTTTTTTTTT   LL
NN       NN   TTTTTTTTTT   LL
NN       NN       TT       LL
NN       NN       TT       LL
NNNN     NN       TT       LL
NNNN     NN       TT       LL
NN  NN   NN       TT       LL
NN  NN   NN       TT       LL
NN    NNNN        TT       LL
NN    NNNN        TT       LL
NN       NN       TT       LL                ....
NN       NN       TT       LL                ....
NN       NN       TT       LLLLLLLLLL        ....
NN       NN       TT       LLLLLLLLLL        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .SBTTL  LOCAL DATA

                                ;
                                ; LOCAL DATA
                                ;

000000                          .PSECT  DATA,D

                                .NLIST  BEX

000000                  K6TMP:  .BLKW   1
000002  000000          NTPT:   .WORD   0

                                ;
                                ; COMPATIBLE SYMBOLS
                                ;
000004                  $PRCLN::
000004                  $PREXT::
000004                  $PRINC::
000004                  $PRLEN::
000004                  $PRLLN::
000004                          .BLKW   1

                                ;
                                ; COMM EXEC FILE NAME
                                ;

                                .IF DF  R$$MPL
                        CEXNAM: .ASCIZ  "RSX11M"
                                .EVEN

                                .IFF
000006  103     105     130     CEXNAM: .ASCIZ  "CEX"
                                .EVEN
                                .ENDC

                                .IF DF  R$$MPL
                        VEXNAM: .ASCIZ  "RSXVEC"
                                .ENDC
                                .EVEN

                                ;
                                ; CETAB FILE NAME
                                ;
000012  103     105     124     CETNAM: .ASCIZ  "CETAB"

                                ;
                                ; SYMBOL TABLE EXTENSION
                                ;
000020  123     124     102     STBEXT: .ASCIZ  "STB"

                                .EVEN


                                ;
                                ; XPT COUNTER BLOCK VALUES
                                ;
```

```
                                        .SBTTL  CENTI - NTINIT PART OF SET SYS + NTINIT

                                ;+
                                ;
                                ; CENTI - PERFORM NTINIT PART OF SET SYS + NTINIT
                                ;
                                ; INPUTS:
                                ;       .PCB = NTPOOL PCB ADDRESS
                                ;       .CCBNM, ETC. = BUF$DF PARAMETERS FROM CETAB
                                ;
                                ; OUTPUTS:
                                ;       NTPOOL ALLOCATION POINTERS INITIALIZED ($QBIAS, $NBIAS, $QSTRT)
                                ;       CCB'S ALLOCATED AND LINKED TO CEX
                                ;       RDB'S AND SDB'S ALLOCATED
                                ;
                                ;-
001354  012700  000000G        CENTI:  MOV     #.CCBNM,R0      ; COPY BUF$DF PARAMETERS INTO CEX
001360  016701  000000G                MOV     CCBNM,R1
        000006                         .REPT   6
                                       MOV     (R0)+,(R1)+     ; CCBNM/SZ, RDBNM/SZ, SDBNM/SZ
                                       .ENDR
001400  016777  000000G 000000G        MOV     .CCBNM,@CCBCT   ; CCBCT
001406  032777  000004  000000G        BIT     #CF.DYN,@CXOPT  ; DYNAMIC CCB ALLOCATION INCLUDED?
001414  001006                        BNE     3$              ; BR IF YES
001416  066777  000000G 000000G        ADD     .RDBNM,@CCBCT   ; ELSE UPDATE CCB COUNT
001424  066777  000000G 000000G        ADD     .RDBNM,@CCBNM   ; ...
001432  016777  000000G 000000G 3$:    MOV     .RDBNM,@RDBCT   ; RDBCT
001440  016777  000000G 000000G        MOV     .SDBNM,@SDBCT   ; SDBCT
001446  016777  000000G 000000G        MOV     .MXPRO,@MAXOV   ; COPY THE MAXIMUM PROTOCOL SIZE ALLOWED
001454  012077  000000G                MOV     (R0)+,@RDBTH

001460  016703  000000G                MOV     .PCB,R3         ; GET NTPOOL PCB ADDRESS
001464  016300  000000G                MOV     P.REL(R3),R0    ; STARTING ADDRESS/BIAS
001470  016303  000000G                MOV     P.BLKS(R3),R3   ; AND LENGTH

001474                                 SWSTK$  30$             ;* ENTER KERNEL MODE

001500  005200                         INC     R0              ;* ADJUST FOR NETLDR HOME BLOCK
001502  005303                         DEC     R3              ;*
001504  005077  000000G                CLR     @NBIAS          ;* ZERO NON-UMR-MAPPED ALLOCATION PTR
001510  005077  000000G                CLR     @PAVL           ;* ASSUME NO FREE SPACE
001514  005077  000000G        5$:     CLR     @XBIAS          ;* ASSUME NO EXTENDED CEX POOL
001520  005077  000000G                CLR     @XAVL           ;* ...
001524  016704  000000G                MOV     .ALLOC,R4       ;* GET SIZE OF EXTENDED CEX POOL
001530  001424                         BEQ     7$              ;* BR IF NO EXTENDED CEX POOL
001532  010046                         MOV     R0,-(SP)        ;* SAVE CURRENT ADDRESS OF NETWORK POOL
001534  060300                         ADD     R3,R0           ;* GET END OF POOL
001536  160400                         SUB     R4,R0           ;* GET START OF EXTENDED CEX POOL
001540  010077  000000G                MOV     R0,@XBIAS       ;* STORE BASE ADDRESS OF EXTENDED POOL
001544  005277  000000G                INC     @XAVL           ;* POINT TO FIRST FREE BLOCK (BLOCK 1)
001550  160403                         SUB     R4,R3           ;* CALCULATE LENGTH OF REMAINING POOL
001552                                 CALL    $MAPX           ;* MAP TO EXTENDED POOL AREA
001556  005022                         CLR     (R2)+           ;* SET UP LISTHEAD - POINTER TO NEXT
001560                                 ASL$    6,R4            ;* CONVERT COUNT TO BYTES
001574  010412                         MOV     R4,(R2)         ;* STORE NUMBER OF BYTES AVAILABLE
001576                                 CALL    @(SP)+          ;* RESTORE PREVIOUS MAPPING
001600  012600                         MOV     (SP)+,R0        ;* RESTORE ADDRESS OF NETWORK POOL
```

NTLCEX CREATED BY MACRO ON 29-JUN-85 AT 00:23 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
E$NBS	000020		#5-62			
E$NCR	000034		#5-62			
E$NCS	000036		#5-62			
E$NIC	000044		#5-62			
E$NLEN	000050		#5-62			
E$NLLA	000012		#5-62			
E$NLNK	000000		#5-62			
E$NML	000040		#5-62			
E$NMR	000024		#5-62			
E$NMS	000030		#5-62			
E$NNOD	000002		#5-62			
E$NRT	000042		#5-62			
E$NRTP	000005		#5-62			
E$NSEG	000010		#5-62			
E$NTIM	000046		#5-62			
E$NUSE	000004		#5-62			
E$STRT	000006		#5-62			
FE.CEX =	******	GX	7-364			
FL.NTI =	******	GX	11-658			
FMASK =	******	GX	7-364			
GTBUF	002250	R	15-908	16-955	#17-978	
HTIM	000024	R	#6-156	*18-1048	18-1071	
IBLK1	000214	R	#6-225	8-417	11-608	
IBLK2	000226	R	6-225	#6-228	8-445	
IBLK3	000240	R	6-228	#6-231		
I$$AS =	******		5-68	7-333	13-755	17-985
KISAR0 =	172340		#5-68			
KISAR6 =	172354		#5-68			
KSAR5 =	******	GX	18-1057	18-1060	18-1076	
K6TMP	000000	R	#6-109			
LF.ACT =	100000		#5-63			
LF.BRO =	000400		#5-63			
LF.BWT =	000007		#5-63			
LF.ENA =	002000		#5-63			
LF.LPB =	001000		#5-63			
LF.MDC =	000100		#5-63			
LF.MFL =	004000		#5-63			
LF.MTP =	000020		#5-63	18-1024		
LF.PAC =	000200		#5-63			
LF.RDY =	040000		#5-63			
LF.REA =	010000		#5-63			
LF.SER =	000040		#5-63			
LF.TIM =	000010		#5-63			
LF.UNL =	020000		#5-63			
LF.X2P =	000000		#5-63			
LGDDB =	******	GX	11-609			
LLCTA =	******	GX	11-605	11-630	11-641	18-1034
LN.CLO =	000000		#5-63			
LN.DUM =	000005		#5-63			
LN.LOA =	000004		#5-63			
LN.LOO =	000003		#5-63			
LN.OAU =	000003		#5-63			

```
                                ;
                                ; SYNTAX ERROR FLAG
                                ;
000356                          SYNERR::.BLKW   1
                                ;
                                ; DATA FOR DTE$DF MACRO
                                ;
000360                          DTEDES::.BLKW   1               ; DTE ADDRESS
000362                          DTEFLG::.BLKW   1               ; FLAGS WORD
                                        DE.ON  == 1             ; SET DTE STATE TO "ON"
                                        DE.OFF == 2             ; SET DTE STATE TO "OFF"
                                        .EVEN
000364                          HSHSZ:: .BLKW   1               ; HASH TABLE SIZE
000366                          HSHADD::.BLKW   1               ; ADDRESS OF HASH TABLE
                                        .EVEN
```

```
                                        .SBTTL  ADD$DF STATE TABLE

                                ;
                                ; LOCAL PHYSICAL STATION ADDRESS (ADD$DF)
                                ;
001402                          STATE$  ADDDF
001402                           TRAN$   %ADD$DF%,,,1,SYNERR
001402                          STATE$
001402                           TRAN$   $NUMBR,$EXIT,AD.NUM
```

NTLCFG CREATED BY MACRO ON 29-JUN-85 AT 00:24 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
ADDDF	000100 R	12-482							
AD.NUM	001534 R	#21-743							
BLKSMN	= 000040	#6-86	27-885						
BLKSMX	= 002000	#6-87	27-885						
CERR	001214 RG	8-173	8-174	8-182	8-183	8-184	8-185	8-186	8-187
		8-188	8-189	#13-518					
CFERR	001224 RG	8-170	8-171	8-172	8-175	8-176	8-177	8-178	8-179
		8-180	8-181	8-190	8-191	8-192	#13-521		
CFGBF	000144 RG	#7-133	11-446	12-490	14-559				
CFGEXT	000006 R	#7-104	10-422						
CFGFLG	000136 RG	#7-121	*9-233	9-331	9-362	9-377	9-387		
CFGKW	000000 RG	12-488	#16-596						
CFGNAM	000000 R	#7-103	10-421						
CFGST	000000 RG	#16-596							
CFGSZ	000352 RG	#7-134	*11-448	12-489	14-560				
CFLIN	001316 RG	8-173	8-174	8-182	8-183	8-184	8-185	8-186	8-187
		8-188	8-189	#14-559					
CHKBLK	002050 RG	#27-880							
CHKWND	002106 RG	#27-894							
CHNLMX	= 007777 G	#6-89							
CTL	000015 RG	#7-110	*9-243	*9-277					
DECPT	= ****** GX	24-810							
DE.OFF	= 000002 G	#7-158							
DE.ON	= 000001 G	#7-157							
DTEDES	000360 RG	#7-155							
DTEFLG	000362 RG	#7-156							
D$SEG	000036	*24-811							
D.NAME	001402 RG	#20-697							
ERRBUF	000027 RG	#7-117	13-522						
ERRMSG	000017 RG	#7-115	13-534						
FG.X2P	= 000001 G	#7-122	9-331						
FG.X3P	= 000002 G	#7-123	9-362	9-377	9-387				
FMT8	001626 RG	8-170	8-171	8-172	8-173	8-174	8-175	8-176	8-177
		8-178	8-179	8-180	8-181	8-186	8-190	8-191	8-192
		#8-201							
FMT8A	001646 RG	#8-202							
FMT8B	001677 RG	8-182	8-183	8-184	8-185	8-187	8-188	8-189	#8-203
FM.8	= 000000	#8-170	#8-171	#8-172	#8-173	#8-174	#8-175	#8-176	#8-177
		#8-178	#8-179	#8-180	#8-181	#8-186	#8-190	#8-191	#8-192
FM.8B	= 000000	#8-182	#8-183	#8-184	#8-185	#8-187	#8-188	#8-189	
FNDPDV	001744 RG	#25-830							
FNLIN	001312 RG	8-172	#14-558						
GETCSR	001622 RG	#23-772							
HSHADD	000366 RG	#7-161							
HSHSZ	000364 RG	#7-160							
IE.EOF	= ****** GX	11-456							
IE.RBG	= ****** GX	11-459							
LF.ACT	= 100000	#5-64							
LF.BRO	= 000400	#5-64							
LF.BWT	= 000007	#5-64							
LF.ENA	= 002000	#5-64							
LF.LPB	= 001000	#5-64							
LF.MDC	= 000100	#5-64							

```
                                ;
                                ; RECORD BUFFER AND BUFFER LENGTH
                                ;
                                CFGBF:: .BLKB   134.
                                CFGSZ:: .BLKW   1

                                ;
                                ; SAVED STACK POINTER FOR ERROR EXIT
                                ;
                                SPSAV:  .BLKW   1

                                ;
                                ; BROADCAST ROUTER ADJACENCIES
                                ;
                                NBRA::  .WORD   0

                                ;
                                ; SYNTAX ERROR FLAG
                                ;
                                SYNERR::.BLKW   1

                                ;
                                ; MARKED-FOR-LOAD FLAG
                                ;
                                MRKFL:: .BLKW   1

                                ;
                                ; POOL.. ALLOCATION VALUES
                                ;
                                XPTLLN::.WORD   0               ; NUMBER OF CIRCUITS
                                ECLLLN::.WORD   0               ; NUMBER OF MAX LOGICAL LINKS
                                ECLNOD::.WORD   0               ; NUMBER OF NODE COUNTERS
                                .BRAD:: .WORD   0               ; NUMBER OF ROUTER ADJACENCIES

                                ;
                                ; ANGLE BRACKETS
                                ;
                                        LBRA='<
                                        RBRA='>

                                        .PSECT
```

```
    744
    745 001252                          STATE$  FILSPC
    746 001252                           TRAN$  <'[>,UIC
    747 001252                           TRAN$  $RAD50,TOK,GETSTR
    748 001252                           TRAN$  $LAMDA
    749
    750 001252                          STATE$  UIC
    751 001252                           TRAN$  $STRNG,,GETSTR
    752
    753 001252                          STATE$
    754 001252                           TRAN$  <',>,FILSPC,GRPUIC
    755 001252                           TRAN$  <']>,FILSPC,USRUIC
    756
    757 001252                          STATE$ TOK
    758 001252                           TRAN$  <':>,FILSPC,DEV
    759 001252                           TRAN$  <']>,FILSPC,USRUIC
    760 001252                           TRAN$  <'.>,EXTEN,FILE
    761 001252                           TRAN$  $LAMDA,$EXIT,FILE
    762
    763 001252                          STATE$  EXTEN
    764 001252                           TRAN$  $STRNG,,GETSTR
    765
    766 001252                          STATE$
    767 001252                           TRAN$  $LAMDA,$EXIT,EXT
    768
```

```
                                ;+
                                ; DEC$DF MACRO STATE TABLE
                                ;-

001252                                STATE$  DECDF
001252                                 TRAN$  $NUMBR,,ECLLNK          ; MAX LOGICAL LINKS

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,NODCT           ; MAX NODE COUNTERS

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,INCT            ; INCOMING TIMER

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,OUTT            ; OUTGOING TIMER

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,INAC            ; INACTIVITY TIMER

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,DELF            ; DELAY FACTOR

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,DELW            ; DELAT WEIGHT

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,IPL             ; INPUT PACKET LIMITER

001252                                STATE$
001252                                 TRAN$  <',>

001252                                STATE$
001252                                 TRAN$  $NUMBR,,LST             ; LINK SERVICE THRESHOLD
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
SF.SVC	= 000002	#5-71
SF.TER	= 001000	#5-73
SF.UNL	= 000040	#5-71
SLTMA	= ****** GX	8-316
SLTNM	= ****** GX	8-318
SPSAV	002446 R	#7-235 *8-299 12-569
STAN	002002 RG	#7-173
STRADD	002232 RG	#7-220
STRPT	002234 RG	#7-221
STRSIZ	002104 RG	#7-202
SYNERR	002452 RG	#7-245 *11-504 *11-507
S$$BAS	= ******	7-107 7-107 7-108 7-108 7-109 7-109 7-110 7-110 7-112
		7-112 7-113 7-113 7-114 7-114 7-117 7-117 7-119 7-119
		7-121 7-121 7-122 7-122 7-127 7-127 7-128 7-128 7-129
		7-129 7-132 7-132 7-133 7-133 7-134 7-134 7-135 7-135
		7-141 7-141 7-142 7-142 7-143 7-143 7-144 7-144 7-145
		7-145 7-146 7-146 7-147 7-147 7-148 7-148 7-149 7-149
		7-150 7-150 7-151 7-151
S.ADD	000004	#5-73
S.CIR	000016	#5-73
S.COST	000001	#5-71
S.DEV	000020	#5-73
S.DIA	000046	#5-73
S.DPA	000032	#5-73
S.DPC	000036	#5-73
S.DUM	000050	#5-73
S.FLAG	000002	#5-73
S.FLG	000000	#5-71
S.HAD	000010	#5-73
S.HST	000006	#5-73
S.LEN	000004	#5-71
S.LENG	000052	#5-73
S.LNK	000000	#5-73
S.LOA	000044	#5-73
S.NMST	000002	#5-71
S.OWNR	000003	#5-71
S.PSS	000022	#5-73
S.SEC	000040	#5-73
S.TER	000042	#5-73
UICBUF	002106 RG	#7-203 7-204
UISZ	= 000014 G	#7-204
XPTLLN	002456 RG	#7-255
ZF.MFL	= ****** GX	8-314
Z.FLG	= ****** GX	*8-314
$ALPHA	= 000022	#17-701
$ANY	= 000020	#17-701
$AZFS	= ****** GX	9-434
$BLANK	= 000006	#17-701
$CBDSG	= ****** GX	15-653
$CFUIC	= ****** GX	9-431
$CLBUF	= ****** GX	*10-459
$CLDEQ	= ****** GX	8-407 12-543

```
                                        .SBTTL  $NTCF3 - SCAN CETAB.MAC FOR PSI COMPONENTS

                                ;+
                                ;
                                ; $NTCF3 - SCAN CONFIGURATION FILE (CETAB.MAC) FOR PSI COMPONENTS
                                ;
                                ; THIS ROUTINE IS CALLED DURING "SET CEX" TO PROCESS THE CONFIG FILE
                                ; FOR NTINIT PARAMETERS PERTAINING TO PSI.
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000000  032767  000000G 000000G $NTCF3::BIT     #FL.NTI,$FLAGS  ; IS THIS SET CEX + NTINIT ?
000006  001475                         BEQ     110$            ; IF EQ, NO
                                ;
000010  010667  000340'                 MOV     SP,SPSAV        ; SAVE SP FOR ERROR EXIT
000014                                  CALL    OPEN            ; OPEN THE CONFIG FILE
000020                          10$:    CALL    READ            ; READ NEXT RECORD
000024  103464                          BCS     100$            ; IF CS, EOF (PSN$DF MISSING - NOTHING TO DO)
000026                                  CALL    $QPSN           ; IS IT PSN$DF ?
000032  103772                         BCS     10$             ; BR IF NO .. KEEP LOOKING
000034                          20$:    CALL    READ            ; READ NEXT RECORD
000040  103462                         BCS     101$            ; IF CS, EOF (X3P$DF MISSING)
000042                                  CALL    $QX3P           ; IS IT X3P$DF?
000046  103772                         BCS     20$             ; BR IF NO .. KEEP LOOKING
                                ;
                                ; READ NEXT RECORD FROM CONFIG FILE
                                ;
000050                          30$:    CALL    READ            ; READ NEXT RECORD
000054  103450                          BCS     100$            ; BR IF EOF
                                ;
                                ; LOOK FOR DTE$DF, DST$DF, RDT$DF OR X29$DF
                                ;
000056                          32$:    CALL    $QDTE           ; IS IT DTE$DF?
000062  103011                          BCC     35$             ; BR IF YES
000064                                  CALL    $QDST           ; IS IT DST$DF?
000070  103023                          BCC     70$             ; BR IF YES
000072                                  CALL    $QRDT           ; IS IT RDT$DF?
000076  103364                          BCC     30$             ; BR IF YES - LOOK FOR NEXT
000100                                  CALL    $QX29           ; IS IT X29$DF?
000104  000761                         BR      30$             ; KEEP LOOKING ANYWAY
                                ;
                                ; PROCESS DTE$DF MACRO AND ASSOCIATED MACROS
                                ;
000106                          35$:    CALL    READ            ; READ NEXT RECORD
000112  103431                          BCS     100$            ; BR IF EOF
000114                                  CALL    $QCHN           ; IS IT CHN$DF?
000120  103372                          BCC     35$             ; E  IF YES - CHECK FOR MORE
000122                                  CALL    $QPVC           ; IS IT PVC$DF?
000126  103367                          BCC     35$             ; BR IF YES - CHECK FOR MORE
000130                                  CALL    $QCUG           ; IS IT CUG$DF?
000134  103364                          BCC     35$             ; BR IF YES - CHECK FOR MORE
000136  000747                         BR      32$             ; ELSE BACK TO TOP LEVEL SCAN
                                ;
```

```
NTLCFP - SCAN CONFIGURATION FIL MACRO V05.03b Saturday 29-Jun-85 00:29  Page 18-2
Symbol table

$QPVC = ****** GX       $QX3P = ****** GX       $TIOUT= ****** GX       .PNUMB= ****** GX       .PSTCN= ****** GX
$QRDT = ****** GX       $REP.N= ****** GX       .PCHAR= ****** GX       .PNUMH= ****** GX       .PSTPT= ****** GX
$QX29 = ****** GX       $SAVAL= ****** GX

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001134    001   (RW,I,LCL,REL,CON)
DATA    001564    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 14155 Words  ( 56 Pages)
Size of core pool: 15496 Words  ( 59 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:27.01
SY:NTLCFP.V2,[132,134]NTLCFP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLCFP
```

NTLCSR CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES			
KSAR7	=	******	GX	6-131			
NOSST		000006	R	#5-72	6-143		
OFFLIN		000120	R	5-82	5-82	#6-159	
PR7	=	******	GX	6-128	6-140		
R$$MPL	=	******		5-58	5-87	6-110	6-145
SST		000000	R	#5-71	6-125		
SSTAB		000020	R	5-71	#5-82		
S.VTTA	=	000002		5-71	5-72		
UAR7SV		000014	R	#5-76	*6-129	6-139	
UDR7SV		000016	R	#5-77	*6-130	6-138	
USAR7	=	******	GX	6-129	6-131	6-139	
USDR7	=	******	GX	6-130	6-132	6-138	
$NLCSR		000000	RG	#6-107			
$$$GLB	=	******		5-71	5-72		

```
                                        .SBTTL  DVSYM - DEFINE SPECIAL DATA STRUCTURE SYMBOLS

                                ;+
                                ; DVSYM - DEFINE SPECIAL DATA STRUCTURE SYMBOLS
                                ;
                                ; THIS ROUTINE IS CALLED TO SCAN THE [131,54]XXXTAB.STB FILE FOR THE FOLLOWING
                                ; SPECIAL SYMBOLS:
                                ;       $XXDCB          DCB ADDRESS
                                ;       $XXEND          END OF DEVICE TABLES
                                ;
                                ; INPUTS:
                                ;       $$DEV=2 CHARACTER DEVICE NAME
                                ;       $NAME=PROCESS NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       DCB=VALUE OF $XXDCB FROM STB FILE
                                ;       LENGTH=DITTO $XXEND
                                ;-
000236  016700  000000G         DVSYM:  MOV     $NAME,R0        ; SETUP DEFAULT FILE NAME
000242  012701  076452'                 MOV     #^RTAB,R1       ; OF XXXTAB
000246                                  CALL    $DFN2           ;       ...
000252  116767  000000G 000005'         MOVB    $$DEV,DNAME+1   ; PREFACE DEVICE NAME WITH "$"
000260  116767  000001G 000006'         MOVB    $$DEV+1,DNAME+2 ;       ...
000266  012700  000004'                 MOV     #DNAME,R0       ; CONVERT IT TO RAD50
000272  010701                          MOV     PC,R1           ;       ...
000274                                  CALL    $CAT5           ;       ...
000300  012702  000126'                 MOV     #DBLK1,R2       ; POINT AT FIRST CONTROL BLOCK
000304  010162  000004         10$:    MOV     R1,4(R2)        ; SETUP SPECIAL NAME
000310  011202                          MOV     (R2),R2         ; GET NEXT CONTROL BLOCK ADDRESS
000312  001374                          BNE     10$             ; IF NE, NOT AT END
000314  010167  000152'                 MOV     R1,DUMMY        ; SETUP DISPATCH TABLE NAME
000320  016767  000166' 000150'         MOV     FOUND,DCB       ; DO STRUCTURES ALREADY EXIST?
000326  000241                          CLC                     ; ASSUME SUCCESS
000330  001056                          BNE     60$             ; IF NE, YES
000332  005000                          CLR     R0              ; SETUP ASCIZ FILE SPEC
000334  005001                          CLR     R1              ; FOR [131,54]XXXTAB.STB
000336  012702  000000'                 MOV     #STBEXT,R2      ;       ...
000342                                  CALL    $AZFS           ;       ...
000346  012700  000126'                 MOV     #DBLK1,R0       ; POINT AT FIRST SYMBOL CONTROL BLOCK
000352                                  CALL    $SYMBL          ; SCAN THE STB FILE
000356  103444                          BCS     10I$            ; IF CS, FILE ERROR
000360  012701  000126'                 MOV     #DBLK1,R1       ; POINT AT FIRST CONTROL BLOCK
000364  005002                          CLR     R2              ; TALLY NUMBER OF ERRORS IN R2
000366  005761  000002                 TST     2(R1)           ; IS THE $XXEND SYMBOL DEFINED?
000372  001007                          BNE     30$             ; IF NE, YES
000374  012767  177777  000136'         MOV     #-1,LENGTH      ; FAKE OUT LENGTH CHECK LATER
000402                          20$:    EMSG$   2A              ; PRINT ERROR MESSAGE
000410  005202                  25$:    INC     R2              ; TALLY THE ERROR
000412  011101                  30$:    MOV     (R1),R1         ; GET NEXT CONTROL BLOCK ADDRESS
000414  001423                          BEQ     50$             ; IF EQ, END OF LIST
000416  005761  000002                 TST     2(R1)           ; IS SYMBOL DEFINED?
000422  001767                          BEQ     20$             ; IF EQ, NO
000424  032761  000001  000010         BIT     #1,10(R1)       ; IS SYMBOL VALUE ODD?
000432  001404                          BEQ     40$             ; IF EQ, NO
000434                                  EMSG$   2N              ; PRINT ERROR MESSAGE
```

B 1
C 1
D 1
E 1 MAPX
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1 NTL
B 2
C 2
D 2
E 2
F 2
G 2
H 2
I 2
J 2 NTLCEX
K 2
L 2
M 2
N 2
B 3
C 3
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5 NTLCFG
I 5

J 5
K 5
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7
N 7
B 8
C 8
D 8
E 8
F 8
G 8 NTLCF2
H 8
I 8
J 8
K 8
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9
J 9
K 9
L 9
M 9
N 9
B 10
C 10
D 10

E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12 NTLCFP
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14 NTLCSR
I 14
J 14
K 14
L 14

M 14
N 14
B 15
C 15 NTLDEV
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16

NTLDEV - BUILD DEVICE DATA STRU MACRO V05.03b Saturday 29-Jun-85 00:30 Page 20-1
Symbol table

```
ALOCB = ****** GX        D.UCBL= ****** GX        L$BMXZ= ****** GX        SLHD  = ****** GX        $ERR2L  000572R     002
A$$CHK= 000000           D.UNIT= ****** GX        L$BSA = ****** GX        STBEXT  000000R     002 $ERR2M  000626R     002
A$$CPS= 000000           ERRBUF  000017R     002  L$$ASG= 000000          S$$WRG= 000000           $ERR2N  000664R     002
A$$PRI= 000000           ERRMSG  000007R     002  L$$DRV= 000000          S$$YSZ= 007600           $ERR2P  000700R     002
A$$TRP= 000000           ETAB    000156R     002  L$$P11= 000001          T4BEND  000174R     002 $ERR2Q  000736R     002
BASE    000170R     002  E$$XPR= 000000           L$$11R= 000000          TS$NHD= ****** GX        $ERR2R  000772R     002
C$$CKP= 000000           FIND    000122R          M$$CRB= 000124          1$$KMG= 000000           $ERR2S  001044R     002
C$$ORE= 000400           FMT4    001404R     002  M$$CRX= 000000          T$$MIN= 000000           $ERR2T  001074R     002
C$$RSH= 177564           FMT4A   001434R     002  M$$FCS= 000000          UNITS   000172R     002 $ERR2U  001146R     002
DBLK1   000126R     002  FMT6    001475R     002  M$$MGE= 000000          U.DCB = ****** GX        $ERR2V  001220R     002
DBLK2   000140R     002  FMT9    001516R     002  M$$NET= 000000          U.RED = ****** GX        $ERR2W  001250R     002
DCB     000150R     002  FM.4  = 000000           M$$OVR= 000000          U.SCB = ****** GX        $ERR2X  001322R     002
DEACB = ****** GX        FM.4A = 000000           N$$ACC= 000001          VAD     001300R          $ERR2Y  001362R     002
DEVHD = ****** GX        FM.6  = 000000           N$$BUF= 000001          V$$CTR= 001000           $FNDEQ  001436R
DNAME   000004R     002  FM.9  = 000000           N$$LDV= 000001          X$$DBT= 000000           $FNOUT= ****** GX
DSKBUF  000000R     003  FNDEA   001412R          N$$MCP= 000001          $AZFS = ****** GX        $IOSB = ****** GX
DUMMY   000152R     002  FOUND   000166R     002  N$$MLL= 000001          $CAT5 = ****** GX        $LBUF = ****** GX
DVALL   000636R          F$$LVL= 000001           N$$MOV= 000010          $CLDEQ= ****** GX        $LLEN = ****** GX
DVDEA   001416R          G$$TPP= 000000           N$$NCT= 000001          $CLIOS= ****** GX        $NAME = ****** GX
DVERR   001324R          G$$TSS= 000000           N$$PEM= 000001          $CLOPE= ****** GX        $NLCDD= ****** GX
DVLBL   000476R          G$$TTK= 000000           P$$P45= 000000          $CUR.N= ****** GX        $NLDEV  000000RG
DVSYM   000236R          G$$WRD= 000000           P$$WRD= 000000          $C5TA = ****** GX        $PCB  = ****** GX
DVXFR   000754R          I$$RAR= 000000           Q$$OPT= 000010          $DFN2 = ****** GX        $PRV.N= ****** GX
D$$BUG= 177514           I$$RDN= 000000           REP4A   001446R     $ERR2A  000176R     002 $READ = ****** GX
D$$ISK= 000000           K$$CNT= 177576           REP4A1  001456R     $ERR2B  000220R     002 $REP.C= ****** GX
D$$L11= 000001           K$$CSR= 177546           REP4A2  001516R     $ERR2C  000256R     002 $REP.P= ****** GX
D$$SYNC= 000000          K$$LDC= 000000           REP9    001600R     $ERR2D  000314R     002 $RLBL = ****** GX
D$$YNM= 000000           K$$TPS= 000074           RVAD    001270R     $ERR2E  000342R     002 $SAVAL= ****** GX
D.DSP = ****** GX        LD$LP = 000000           R$$DER= 000000          $ERR2F  000370R     002 $SYMBL= ****** GX
D.NAM = ****** GX        LENGTH  000136R     002  R$$K11= 000001          $ERR2H  000426R     002 $TIOUT= ****** GX
D.PCB = ****** GX        L$BFLG= ****** GX        R$$SND= 000000          $ERR2J  000456R     002 $$DEV = ****** GX
D.UCB = ****** GX        L$BLDZ= ****** GX        R$$11M= 000000          $ERR2K  000530R     002 $$TBL = ****** GX


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001624    001   (RW,I,LCL,REL,CON)
DATA    001544    002   (RW,D,LCL,REL,CON)
..BUF   000002    003   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10301 Words  ( 41 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:29.54
SY:NTLDEV.V2,[132,134]NTLDEV/CR/-SP-SY:[1,1]RSXMCM.SML/ML [130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLDEV
```

NTLGET -
PUTAST -

```
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
```


NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

NTLGET -
Symbol t

```
                                        .SBTTL  PUTAST - PUT COMPLETION AST IN CALLER'S AST QUEUE

                                ;+
                                ; PUTAST - PUT A COMPLETION AST IN THE CALLER TASK'S AST QUEUE
                                ;
                                ; INPUTS:
                                ;       $$TASK=CALLING TASK'S NAME
                                ;       $FUNC=SUCCESS CODE
                                ;       $$AST=AST ADDRESS IN TASK
                                ;       $$PRM=PARAMETER TO RETURN TO TASK
                                ;
                                ; OUTPUTS:
                                ;       AN AST IS QUEUED FOR THE CALLER.
                                ;-
                                        .IF NDF I$$AS
000320                          PUTAST:
000320                                  SWSTK$  30$                     ;; ENTER SYSTEM STATE
000324  012703  000000G                 MOV     #$$TASK,R3              ;; POINT TO TASK NAME
000330                                  CALL    @$RSTD                  ;; FIND CALLER'S TCB ADDRESS
000334  103406                          BCS     10$                     ;; IF CS, NOT THERE - TOO BAD
000336  010005                          MOV     R0,R5                   ;; ELSE, COPY TCB ADDRESS
000340  012701  000016                  MOV     #7*2,R1                 ;; ELSE, SET AST BLOCK LENGTH
000344                                  CALL    @ALOCB                  ;; ALLOCATE SOME POOL
000350  103006                          BCC     20$                     ;; IF CC, GOT IT - PROCEED
000352                          10$:    RETC    R0                      ;; SET USER C-BIT
000364  000421                          BR      30$                     ;; ... AND RETURN

000366  010002                  20$:    MOV     R0,R2                   ;; COPY ADDRESS OF BLOCK ALLOCATED
000370  005720                          TST     (R0)+                   ;; BUMP OVER LINK WORD
000372  010120                          MOV     R1,(R0)+                ;; SET LENGTH OF BLOCK (FOR DEALLOCATION)
000374  012720  000022                  MOV     #11*2,(R0)+             ;; SET STACK SPACE TO RESERVE FOR TASK
000400  016720  000000G                 MOV     $$AST,(R0)+             ;; SET AST VIRTUAL ADDRESS FOR TASK
000404  012720  000002                  MOV     #2,(R0)+                ;; SET PARAMETER COUNT
000410  016720  000000G                 MOV     $$PRM,(R0)+             ;; SET PARAMETER #1 (USER SUPPLIED)
000414  016710  000000G                 MOV     $FUNC,(R0)              ;; SET PARAMETER #2 (COMPLETION STATUS)
000420  010500                          MOV     R5,R0                   ;; POINT TO TASK TCB
000422  010201                          MOV     R2,R1                   ;; POINT TO AST BLOCK
000424                                  CALL    @@ASTT                  ;; QUEUE THE AST TO THE CALLING TASK
000430                          30$:    RETURN                          ;; BACK TO USER STATE AND RETURN
                                        .ENDC

000001                                  .END
```

```
                                        .SBTTL  KMC LOADER (PROCESS LOAD)

                                ;+
                                ; $KMC - LOAD MICRO-CODE FOR INDIVIDUAL PROCESS
                                ;
                                ; INPUTS:
                                ;       $PDVA  = PDV ADDRESS FOR DDM PROCESS
                                ;       $NAME  = RAD50 PDV/DEVICE NAME
                                ;       $SLTA  = SYSTEM LINE TABLE ENTRY ADDRESS
                                ;       $$CSR  = KMC CSR ADDRESS
                                ;       $NTLPT = ADDRESS OF NTL CONTROL DATA BLOCK
                                ;       .CXKMC IN DATA BLOCK CONTAINS ADDRESS OF FIRST LOAD LIST BLOCK
                                ;         (CONTAINS ZERO IF NO BLOCKS ALLOCATED YET)
                                ;       NO REGISTER SAVING IS NECESSARY
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000000                          $KMC::
000000  016700  000000G                MOV     $SLTA,R0                ; GET THE SLT ADDRESS
000004  005760  000004                TST     L.DDS(R0)               ; IS THIS THE FIRST LINE TO BE LOADED ?
000010  001011                        BNE     10$                     ; NO .. NOTHING TO DO
                                ;
                                ; KMC WAS NOT PREVIOUSLY IN LOAD LIST, MUST TRY LOADING IT
                                ;
000012  016700  000000G               MOV     $NAME,R0                ;B-GET PDV/DEVICE NAME
000016  016704  000000G               MOV     $$CSR,R4                ;B-GET KMC CSR ADDRESS
000022                                CALL    KMCLD                   ;B-LOAD THE KMC WITH MICRO-CODE
000026  103402                        BCS     10$                     ;B-BR IF LOAD WAS BAD
000030                                CALL    KMCLL                   ; HOUSEKEEP THE LIST OF LOADED KMC'S

000034                          10$:    RETURN
```

```
                                .SBTTL  FILE ERROR SUB-FORMATTER

                        ;+
                        ; -KMCFEC- FILE CLOSED, REPLACE FILNAM AND ERROR CODE (MAYBE) WHEN FORMATTING
                        ; -KMCFEO- FILE OPEN, REPLACE FILNAM AND ERROR CODE (MAYBE) WHEN FORMATTING
                        ;
                        ; THIS ROUTINE WILL REPLACE THE ASTERISK IN THE PREFIX STRING WITH
                        ; THE FILE NAME, AND THE ASTERISK IN THE SUFFIX STRING WITH THE
                        ; ERROR CODE, COPYING BOTH IN THE MESSAGE BUFFER.  IF THERE IS NO
                        ; ASTERISK IN THE SUFFIX STRING, IT IS ASSUMED THAT NO ERROR CODE
                        ; IS NEEDED.  THERE MUST BE AN ASTERISK IN THE PREFIX STRING.
                        ;
                        ; INPUTS:
                        ;       R3 = ADDRESS OF MESSAGE PREFIX STRING (FORMAT STRING)
                        ;       R4 = ADDRESS OF MESSAGE BUFFER
                        ;       R5 = ADDRESS OF 4TH WORD IN ERROR MESSAGE CONTROL BLOCK
                        ;       FILE IS CURRENTLY OPEN VIA $CLOPE
                        ;       ERROR CODE (IF ANY) IS CONTAINED IN BYTE @$CLIOS
                        ;
                        ; OUTPUTS:
                        ;       R0, R1, AND R2 ARE DESTROYED
                        ;       R3 = ADDRESS AFTER END OF SUFFIX STRING (AFTER NULL)
                        ;       R4 = ADDRESS AT END OF MESSAGE (AT NULL)
                        ;-
                                .ENABL  LSB

001312                  KMCFEC: CALL    $REP.P                  ;USE TO COPY UP TO END OF FILE NAME
001316  000402                  BR      10$                     ;REST IS ALL COMMON CODE

001320                  KMCFEO: CALL    $REP.C                  ;USE TO COPY UP TO END OF FILE NAME

001324  112324          10$:    MOVB    (R3)+,(R4)+             ;COPY REST OF PREFIX
001326  001376                  BNE     10$                     ;BR IF NOT DONE
001330  010503                  MOV     R5,R3                   ;COPY ADDRESS OF WORD BEFORE SUFFIX
001332  005723                  TST     (R3)+                   ;SKIP A WORD
001334  112314          20$:    MOVB    (R3)+,(R4)              ;MOVE BYTES TO FINAL MESSAGE
001336  001414                  BEQ     30$                     ;BR IF DONE
001340  122724  000052          CMPB    #'*,(R4)+               ;JUST MOVED AN ASTERISK?
001344  001373                  BNE     20$                     ;BR IF NO, KEEP GOING
001346  005304                  DEC     R4                      ;BACK UP OVER THE ASTERISK
001350  010400                  MOV     R4,R0                   ;COPY BUFFER POINTER
001352  117701  000000G         MOVB    @$CLIOS,R1              ;GET ERROR CODE
001356  005002                  CLR     R2                      ;SUPPRESS LEADING ZEROES
001360                          CALL    $CBDSG                  ;CONVERT BINARY TO SIGNED DECIMAL
001364  010004                  MOV     R0,R4                   ;GET ADDRESS AFTER ERROR CODE
001366  000762                  BR      20$                     ;KEEP MOVING BYTES
001370                  30$:    RETURN                          ;DONE

                                .DSABL  LSB
```

```
                                        .SBTTL  LOCAL DATA
                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

000000  000000                 TEMP:   .WORD   0               ; TEMP LOCATION

                                        .NLIST  BEX

000002                          XPTPDV: .BLKW   1               ; PDV ADDRESS FOR XPT
000004                          NODCTR: .BLKW   1               ; LOCAL COPY OF NUMBER OF NODE COUNTERS

                                ;
                                ; SYMBOL TABLE EXTENSION
                                ;
000006  123     124     102     STBEXT: .ASCIZ  "STB"

                                ;
                                ; ASCII DISPATCH TABLE NAME
                                ;
000012  044     130     130     DSPTAB: .ASCII  "$XXXTB"

                                        .EVEN

                                        .IF DF  R$$MPL

                                ;
                                ; PROCESS VECTOR TABLE SYMBOLS
                                ;
                                VECTAB: .ASCII  "$XXX"
                                        .BYTE   40
                                        .BYTE   40
                                VECLEN: .ASCII  "$XXX"
                                        .BYTE   40
                                        .BYTE   40

                                ;
                                ; CETAB FILE NAME
                                ;
                                CETNAM: .ASCIZ  /CETAB/
                                        .EVEN
                                ;
                                ; CONTROLLER TABLE SYMBOL
                                ;
                                CTBLK:  .WORD   0,0
                                CTBSYM: .RAD50  "$CTXXX"
                                CTVAL:  .BLKW   1

                                        .ENDC

                                ;
                                ; PROCESS DISPATCH TABLE SYMBOL
                                ;

                                        .IF DF  R$$MPL
```

```
                                        .SBTTL  $ALECL - ALLOCATE ECL NODE COUNTERS
                                ;+
                                ; $ALECL - ALLOCATE ECL NODE COUNTERS FROM POOL..
                                ;
                                ; INPUTS:
                                ;       $PDVA - ECL'S PDV ADDRESS
                                ;       $NN - MAXIMUM NODE ADDRESS
                                ;       D$MNOD OF HOME BLOCK - NUMBER OF NODE COUNTERS TO ALLOCATE
                                ;
                                ; OUTPUTS:
                                ;       N$ENC     (ECL'S DDB) - APR6 VIRTUAL ADDRESS OF START OF NODE COUNTERS
                                ;       N$ENC+2   (ECL'S DDB) - APR6 VIRTUAL ADDRESS OF END OF NODE COUNTERS
                                ;       N$ENC+4   (ECL'S DDB) - APR BIAS FOR NODE COUNTERS
                                ;
                                ;-
001220                          $ALECL::CALL    $SAVAL                  ; SAVE ALL REGISTERS
001224  017701  000000G                 MOV     @#ECPT,R1               ; GET NUMBER OF NODE COUNTERS
001230  005000                          CLR     R0                      ; ...
001232  156100  000041                  BISB    D$MNOD(R1),R0           ; ...
001236  010067  000000G                 MOV     R0,$$NODC               ; ...
001242  012701  000050                  MOV     #E$NLEN,R1              ; GET LENGTH OF ECL COUNTERS
001246                                  CALL    $MUL                    ; CALCULATE NUMBER OF BYTES
001252  062701  000077                  ADD     #77,R1                  ; ROUND TO BLOCK BYTE BOUNDARY
001256  042701  000077                  BIC     #77,R1                  ; ...
                                
001262                                  ASR$    6,R1                    ; CONVERT TO BLOCK COUNT
001276  001465                          BEQ     40$                     ; BR IF NONE TO ALLOCATE
001300                                  CALL    $ALN18                  ; ALLOCATE COUNTERS FROM NON-UMR SPACE
001304  103463                          BCS     101$                    ; BR IF ERROR
001306  016705  000000G                 MOV     $PDVA,R5                ; GET ECL'S PDV ADDRESS
001312  016595  000000G                 MOV     Z.DAT(R5),R5            ; GET ECL'S DDB ADDRESS
001316  010065  000046                  MOV     R0,N$ENC+4(R5)          ; STORE APR BIAS NODE COUNTERS
001322  016767  0000000G 000004'        MOV     $$NODC,NODCTR           ; MAKE LOCAL COPY OF NO. NODE COUNTER
                                
001330                                  SWSTK$  40$                     ;; ENTER SYSTEM STATE
001334                                  CALL    $MAPX                   ;; MAP TO COUNTERS
001340  005001                          CLR     R1                      ;; POINT TO START OF COUNTERS
001342  050201                          BIS     R2,R1                   ;; SET CORRECT APR BITS
001344  005004                          CLR     R4                      ;; POINT TO START OF COUNTERS
001346                                  BIAS    R4                      ;; ... (ECL NEEDS APR6 BIAS)
001356  010465  000042                  MOV     R4,N$ENC(R5)            ;; STORE START OF COUNTERS
001362  016746  000004'                 MOV     NODCTR,-(SP)            ;; GET NUMBER OF NODE COUNTERS
001366  010102                          MOV     R1,R2                   ;; POINT TO START OF COUNTERS
001370  062702  000050          20$:    ADD     #E$NLEN,R2              ;; POINT TO NEXT COUNTER BLOCK IN LIST
001374  010204                          MOV     R2,R4                   ;; COPY ADDRESS OF NEXT BLOCK
001376                                  BIAS    R4                      ;; SET CORRECT APR BIAS FOR ECL
001406  010421                          MOV     R4,(R1)+                ;; STORE LINK WORD
001410  012703  000023                  MOV     #<<E$NLEN-2>/2>,R3      ;; GET NUMBER OF WORDS TO ZERO
001414  005021                  30$:    CLR     (R1)+                   ;; ZERO THE BLOCK
001416                                  SOB     R3,30$                  ;; ...
001422  005316                          DEC     (SP)                    ;; MORE TO ALLOCATE?
001424  003361                          BGT     20$                     ;; BR IF YES
001426  005726                          TST     (SP)+                   ;; CLEAN UP STACK
001430  162701  000050                  SUB     #E$NLEN,R1              ;; POINT TO START OF PREVIOUS BLOCK
001434  005011                          CLR     (R1)                    ;; INDICATE END OF LIST
```

```
                                .SBTTL  MACRO DEFINITIONS

                        ;
                        ; LIBRARY MACROS
                        ;
                                .MCALL  SLTDF$,CCBDF$

000000                          SLTDF$                  ; DEFINE SLT OFFSETS AND SYMBOLS
000000                          CCBDF$                  ; DEFINE PROCESS DIPATCH OFFSETS

                                .IF DF  R$$MPL
                                .MCALL  GIN$,DIR$
                                .ENDC
```

```
                                        .SBTTL  $SLIN - SET LINE

                                ;+
                                ; $SLIN - "SET LINE"
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;       $SLN=SYSTEM LINE NUMBER
                                ;       $$CSR=CSR ADDRESS
                                ;       $$PRI=INTERRUPT PRIORITY (1 THRU 7)
                                ;       $$VECT=INTERRUPT VECTOR ADDRESS
                                ;       $$MTP=STATION COUNT (MULTIPOINT ONLY)
                                ;       $$SNUM=STATION NUMBER TABLE (MULTIPOINT ONLY)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;
                                ;-
000314                          $SLIN:
000314  016701  000140'                MOV     $DLCPV,R1       ; GET DLC PDV ADDRESS
000320  001424                         BEQ     20$             ; IF EQ, COMBINATION DDM/DLC
000322  004767  000146                 CALL    AIC             ; IS DLC ALREADY IN CORE?
000326  001016                         BNE     10$             ; IF NE, YES
000330  012767  003000  000014'        MOV     #MS.INC!MS.EXT,$MISS    ; LOOK FOR PROCESS INCREMENT
000336  004767  000000G                CALL    $NLCFG          ; ..IN CETAB
000342  103451                         BCS     50$             ; IF CS, FAILED
000344  004767  000000G                CALL    $XFER           ; TRANSFER DLC INTO CORE
000350  103446                         BCS     50$             ; IF CS, FAILURE (NOTHING TO CLEAN UP)
000352  052767  010000  001232'        BIS     #FL.PX1,$FLAGS  ; INDICATE SUCCESSFUL DLC TRANSFER
                                
                                        .IF DF  R$$MPL
                                        CALL    VECSET          ; SETUP VECTOR TABLE
                                        .ENDC
                                
000360  010160  000000G                MOV     R1,Z.PCB(R0)    ; SETUP PCB ADDRESS IN PDV
000364  004767  000000G        10$:    CALL    $IMP            ; SETUP DLC LINE TABLE
000370  103437                         BCS     101$            ; IF CS, FAILURE
000372  016701  000142'        20$:    MOV     $DDMPV,R1       ; GET DDM PDV ADDRESS
000376  004767  000072                 CALL    AIC             ; IS DDM ALREADY IN CORE?
000402  001016                         BNE     25$             ; IF NE, YES
000404  012767  003000  000014'        MOV     #MS.INC!MS.EXT,$MISS    ; LOOK FOR PROCESS INCREMENT
000412  004767  000000G                CALL    $NLCFG          ; ..IN CETAB
000416  103424                         BCS     101$            ; IF CS, FAILED
000420  004767  000000G                CALL    $XFER           ; TRANSFER DDM INTO CORE
000424  103421                         BCS     101$            ; IF CS, FAILURE
000426  052767  020000  001232'        BIS     #FL.PX2,$FLAGS  ; INDICATE SUCCESSFUL DDM TRANSFER

                                        .IF DF  R$$MPL
                                        CALL    VECSET          ; SETUP VECTOR TABLE
                                        .ENDC

000434  010160  000000G                MOV     R1,Z.PCB(R0)    ; SETUP PCB ADDRESS IN PDV
000440  004767  000000G        25$:    CALL    $CTSYM          ; SETUP CTB ON RSX-11M+
000444  103411                         BCS     101$            ; IF CS, FATAL ERROR
```

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 6

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$MAXOV	= ****** GX	5-143
$MISS	000014 RG	#6-228 7-477 *11-674 *11-691 *14-818 *14-830
$MXVCT	000076 RG	#5-138
$NAME	000064 RG	#6-293 *12-739
$NBIAS	= ****** GX	5-142
$NETPF	= ****** GX	6-181
$NLCFG	= ****** GX	7-479 11-675 11-692 14-819 14-831
$NLDEV	= ****** GX	8-543
$NLGET	= ****** GX	7-472
$NLINI	= ****** GX	7-471
$NLPUT	= ****** GX	7-486
$NLRQB	= ****** GX	7-473
$NONSI	= ****** GX	5-119
$NSO	= ****** GX	6-171
$NS1	= ****** GX	6-172
$NS2	= ****** GX	6-173
$NS3	= ****** GX	6-174
$NS4	= ****** GX	6-175
$NS5	= ****** GX	6-176
$NS6	= ****** GX	6-177
$NS7	= ****** GX	6-178
$NTCFS	= ****** GX	13-788 13-790
$NTCF3	= ****** GX	13-783 13-785
$NTICF	= ****** GX	13-774 13-778
$NTL	000000 R	#7-455 14-835
$NTLPT	= ****** GX	6-193
$NTUIC	= ****** GX	5-137
$OBJHD	= ****** GX	6-194
$PARHD	= ****** GX	5-122
$PCB	000070 RG	#6-295 *8-507 8-556 *12-753
$PDVA	000062 RG	#6-292 6-340 8-506 8-518 8-526 8-546 8-554 *12-734 14-815
$PDVID	= ****** GX	6-179
$PDVNM	= ****** GX	6-165
$PDVTA	= ****** GX	5-158
$PSIPT	= ****** GX	6-213
$PUMR	= ****** GX	6-195
$QASTT	= ****** GX	5-114
$QBIAS	= ****** GX	6-205
$QSTRT	= ****** GX	6-206
$RCDEV	000024 RG	#6-270
$RDBAF	= ****** GX	5-152
$RDBCT	= ****** GX	5-147
$RDBLH	= ****** GX	6-190
$RDBNM	= ****** GX	6-207
$RDBS7	= ****** GX	6-187
$RDBTH	= ****** GX	5-144
$RSXNM	000000 RG	#6-223
$SCEX	000552 R	6-404 #13-773
$SCIR	000636 R	6-411 #14-815
$SDBAF	= ****** GX	5-154
$SDBCT	= ****** GX	5-146
$SDBLH	= ****** GX	6-191

NTLRQB
$NLRQB

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241

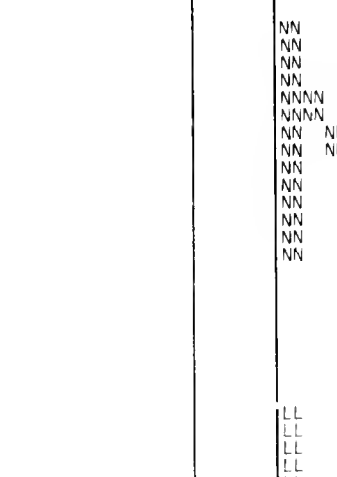
NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 7

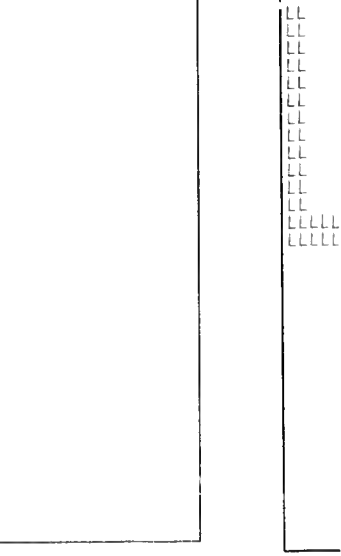
NTLRQB
SCEX -

```
                                        .SBTTL  $NLRQB - PROCESS RECEIVE DATA BLOCK

                                ;+
                                ; $NLRQB - PROCESS RECEIVE DATA BLOCK
                                ;
                                ; THIS ROUTINE IS CALLED TO PROCESS THE NETLDR REQUEST BLOCK AND TO
                                ; INITIALIZE THE LOCATIONS "$FUNC" AND "$MISS".
                                ;
                                ; INPUTS:
                                ;       $$LRB=LOAD REQUEST BLOCK
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       ALL REGISTERS=DESTROYED
                                ;       $FUNC=FUNCTION CODE (0=NOP,
                                ;       $MISS=MISSING PARAMETERS BIT MASK
                                ;       $PDVA=PDV ADDRESS (SET/CLEAR LINE/PROCESS ONLY)
                                ;       $SLTA=SLT ADDRESS (SET/CLEAR LINE ONLY)
                                ;
                                ;-
000000                          $NLRQB::
000000  010667  000132'                MOV     SP,SPSAV        ; SAVE STACK POINTER
000004  012703  000004G                MOV     #$$LRB+4,R3     ; GET REQUEST BLOCK ADDRESS
000010  012304                         MOV     (R3)+,R4        ; GET FUNCTION CODE
000012  042704  177400                 BIC     #^C<377>,R4     ;       ...
000016  001421                         BEQ     101$            ; IF ZERO, ILLEGAL FUNCTION
000020  022704  000010                 CMP     #FMAX,R4        ; IS IT A VALID FUNCTION CODE?
000024  103416                         BLO     101$            ; IF LO, NO
000026  010405                         MOV     R4,R5           ; COPY FUNCTION CODE
000030  006305                         ASL     R5              ; MAKE A WORD INDEX
000032  016505  177776'                MOV     FTAB-2(R5),R5   ; GET SUBROUTINE TABLE POINTER
000036  001411                         BEQ     101$            ; IF ZERO, ILLEGAL FUNCTION
000040  005067  000000G                CLR     $MISS           ; NO PARAMETERS MISSING YET (CLEARS C-BIT)
000044                          10$:    CALL    @(R5)+          ; KEEP CALLING SUBROUTINES
000046  103402                         BCS     20$             ; UNTIL C-BIT IS SET
000050  005715                         TST     (R5)            ; OR END OF TABLE (0) IS REACHED
000052  001374                         BNE     10$             ;       ...
000054  010467  000000G          20$:  MOV     R4,$FUNC        ; SET FUNCTION CODE
000060                                 RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000062                          101$:   FMSG$R  99              ; ILLEGAL FUNCTION CODE
```

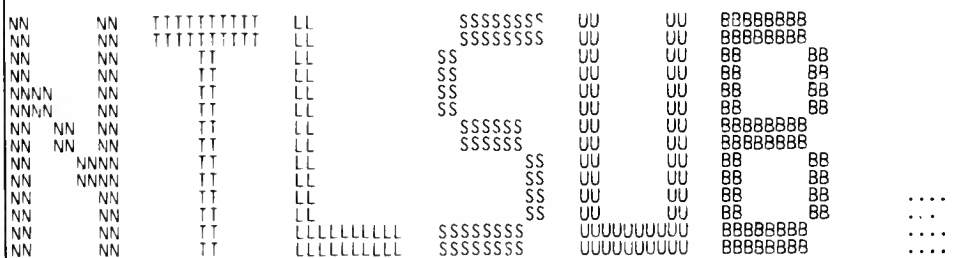
```
001434  004267  000000G                 JSR     R2,$SAVVR       ; SAVE R0,R1,R2 (CO-ROUTINE)
001440  014101                           MOV     -(R1),R1        ; GET SYSTEM LINE NUMBER (BYTE)
001442  042701  177400                   BIC     #^C<377>,R1     ;    ...
001446  006301                           ASL     R1              ; CONVERT TO A WORD INDEX
001450  060166  000012                   ADD     R1,10.(SP)      ; ADJUST SLT POINTER
001454  017666  000012  000012           MOV     @10.(SP),10.(SP) ; POINT AT SLT
001462  062766  000000  000012           ADD     #L.FLG,10.(SP)  ; POINT AT FLAGS WORD
001470                                   RETURN                  ; RESTORE R0,R1,R2; RETURN TO 25$

001472  032776  000400  000000  25$:     BIT     #LF.BRO,@(SP)   ; BROADCAST CIRCUIT ?
001500  001421                           BEQ     27$             ; IF NO - BRANCH
001502  062716  000022                   ADD     #<L.MPF-L.FLG>,(SP) ; IF YES - POINT AT START OF TRIB FLAGS

001506  014103                           MOV     -(R1),R3        ; GET TRIB NUMBER
001510  005721                           TST     (R1)+           ; RESTORE R1
001512  000303                           SWAB    R3              ;    ...
001514  042703  177400                   BIC     #^C<377>,R3     ;    ...
001520  006303                           ASL     R3              ; GET TRIB INDEX
001522  006303                           ASL     R3              ;    ...
001524  060316                           ADD     R3,(SP)         ; POINT TO TRIB FLAG

001526  132736  000200                   BITB    #SF.ACT,@(SP)+  ; IS CHANNEL ACTIVE ?
001532  001412                           BEQ     30$             ; IF NO - BRANCH
001534                                   EMSG$   Z3              ; IF YES - PRINT ERROR
001542  000406                           BR      30$             ; CONTINUE

001544  032736  100000          27$:     BIT     #LF.ACT,@(SP)+  ; IS CHANNEL 1-N ACTIVE?
001550  001403                           BEQ     30$             ; IF EQ, NO
001552                                   EMSG$   Z3              ; CHANNEL STILL ACTIVE (SETS R0 NON-ZERO)
001560  020102                  30$:     CMP     R1,R2           ; SCAN ENTIRE CHANNEL TABLE
001562  001315                           BNE     20$             ;       ...
001564                          40$:

001564  005400                           NEG     R0              ; ANY ERRORS?
001566  103531                           BCS     150$            ; IF CS, YES
001570  005001                           CLR     R1              ; INDICATE NO DEVICE ERROR
001572                                   SWSTK$  90$             ;* ENTER KERNEL MODE
001576  016701  000000G                 MOV     $PDVA,R1        ;* GET PDV ADDRESS
001602  016702  000000G                 MOV     DEVHD,R2        ;* AND RSX11M DEVICE POINTER
001606  011202                  50$:     MOV     (R2),R2         ;* GET NEXT DEVICE IN LIST
001610  001446                           BEQ     80$             ;* IF EQ, ALL DONE
001612  026162  000000G 000000G           CMP     Z.PCB(R1),D.PCB(R2) ;* IS OUR LLC ALSO THIS DEVICE?
001620  001372                           BNE     50$             ;* IF NE, NO
001622  016203  000000G                 MOV     D.UCB(R2),R3    ;* GET FIRST UCB ADDRESS
001626  116204  000000G                 MOVB    D.UNIT(R2),R4   ;* AND FIRST UNIT NUMBER
001632  032763  000000G 000000G 60$:     BIT     #DV.MNT,U.CW1(R3) ;* MOUNTABLE DEVICE?
001640  001010                           BNE     65$             ;* IF NE, YES
001642  132763  000000G 000000G           BITB    #US.BSY,U.STS(R3) ;* IS UNIT STILL BUSY?
001650  001414                           BEQ     70$             ;* IF EQ, NO
001652  016266  000000G 000004           MOV     D.NAM(R2),4(SP) ;* SAVE NAME OF BUSY DEVICE
001660  000422                           BR      80$             ;*
001662  132763  000000G 000000G 65$:     BITB    #US.MNT,U.STS(R3) ;* IS UNIT MOUNTED?
001670  001004                           BNE     70$             ;* IF NE, NO
001672  016266  000000G 000002           MOV     D.NAM(R2),2(SP) ;* SAVE NAME OF MOUNTED DEVICE
001700  000412                           BR      80$             ;*
001702  066203  000000G          70$:     ADD     D.UCBL(R2),R3   ;* GET NEXT UCB ADDRESS
001706  105204                           INCB    R4              ;* AND NEXT UNIT NUMBER
```

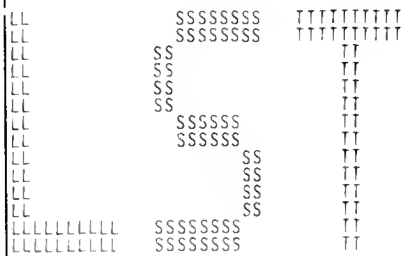
```
                                        .SBTTL  REP7 - REPLACE REQUESTING TASK'S NAME
                                ;+
                                ; REP7 - REPLACE REQUESTING TASK'S NAME
                                ;
                                ; NOTE:
                                ;       THIS ROUTINE HAS BEEN CHANGED INTO A NOP
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-
003420                          REP7:   RETURN
```



FILEID**NTLSUB





NTLSUB - COMMON SUBROUTINES MACRO V05.03b Saturday 29-Jun-85 00:33 Page 4

NTLSUB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
DFEXT	000121 R	#5-141 7-215
D.DSP	= ****** GX	12-366
D.UCBL	= ****** GX	12-377
FILE	000243 R	#5-153 7-199 7-224
FNERR	000233 R	#5-151 10-311
FNPTR	000030 R	#5-92 *7-219 10-312
FO.WBT	= ****** GX	5-87
LDPWF	= ****** GX	12-368
M$$MGE	= 000000	5-98 5-125 12-357
NAMES	000125 R	#5-146 11-337
Q.IOFN	= 000002	5-87
Q.IOPL	= 000014	*10-315 *10-316
R$$MPL	= ******	5-61 5-76 12-370 13-399
R$$11M	= 000000	5-97 5-113 5-124 12-357
SLFNP	000222 R	7-221 8-260 #11-329
TIOUT	000300 R	#5-87 *10-315 *10-316 10-317
UC.PWF	= ****** GX	12-362
U.CTL	= ****** GX	12-362
U.CW3	= ****** GX	*12-376
U.SCB	= ****** GX	12-367
$AZFS	000042 RG	#7-197
$CBDSG	= ****** GX	9-293
$CFUIC	000052 RG	#5-113
$CLEVL	= ****** GX	11-330
$CLIOS	= ****** GX	9-291
$CUR.N	000216 RG	#9-281
$CXUIC	000072 RG	#5-125
$C5TA	= ****** GX	6-178
$DFN	000000 RG	#6-169
$DFNAM	000112 RG	#5-136 6-174 7-208
$DFN2	000002 RG	#6-171
$DFUIC	000032 RG	#5-97 7-202
$FEATR	= ****** GX	12-376
$FNOUT	000270 RG	#10-310
$NLCDD	000352 RG	#12-359
$PRV.N	000224 RG	#9-284
$REP.C	000160 RG	#8-251 9-282
$REP.N	000230 RG	9-283 #9-286
$REP.P	000164 RG	#8-254 9-285
$TIOUT	000302 RG	#10-314
$$$ARG	= 000003	#5-87 5-87 #5-87 5-87 #5-87 5-87 #5-87 5-87 5-87
$$$GLB	= ******	5-87

NTLSUB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 2

**FILE

NN
NN
NN
NN
NNNN
NNNN
NN NN
NN NN
NN
NN
NN
NN
NN
NN

LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LL
LLLLLL
LLLLLL

NTLTMP

FILEID**NTLTMP

```
NN       NN   TTTTTTTTTT   LL           TTTTTTTTTT   MM       MM   PPPPPPPP
NN       NN   TTTTTTTTTT   LL           TTTTTTTTTT   MM       MM   PPPPPPPPP
NN       NN       TT       LL               TT       MMMM   MMMM   PP      PP
NN       NN       TT       LL               TT       MMMM   MMMM   PP      PP
NNNN     NN       TT       LL               TT       MM  MM  MM   PP      PP
NNNN     NN       TT       LL               TT       MM  MM  MM   PP      PP
NN  NN   NN       TT       LL               TT       MM       MM   PPPPPPPP
NN  NN   NN       TT       LL               TT       MM       MM   PPPPPPPP
NN    NNNN       TT       LL               TT       MM       MM   PP
NN    NNNN       TT       LL               TT       MM       MM   PP
NN       NN       TT       LL               TT       MM       MM   PP              ....
NN       NN       TT       LL               TT       MM       MM   PP              ....
NN       NN       TT       LLLLLLLLLL       TT       MM       MM   PP              ....
NN       NN       TT       LLLLLLLLLL       TT       MM       MM   PP              ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```


NTLTMP - TEMPLATE PROCESSING MACRO V05.03b Saturday 29-Jun-85 00:34 Page 4

NTLTMP

```
                                ;+
                                ; $VFSPC - VERIFY SPECIAL OPERATOR
                                ; $VFLV1 - VERIFY PRIMARY TEMPLATE (LEVEL 1)
                                ;
                                ; INPUTS:
                                ;       R0=$FLAG1 BIT CODE FOR THE OPERATOR ($VFSPC ONLY)
                                ;
                                ; OUTPUTS:
                                ;       OPERATOR BIT SET IN $FLAG1
                                ;       ERROR IF OPERATOR HAS ALREADY BEEN SEEN OR IF NOT AT LEVEL 1
                                ;-

                                        .ENABL  LSB

000562                          $VFSPC::
000562  030067  000000'                 BIT     R0,$FLAG1       ; IT THIS THE FIRST TIME FOR THIS GUY?
000566  001007                          BNE     101$            ; IF NE, NO
000570  050067  000000'                 BIS     R0,$FLAG1       ; INDICATE OPERATOR SEEN
000574                          $VFLV1::
000574  026727  000032' 000034'         CMP     $SZPTR,#$SZPTR+2 ; IS THIS THE TOP LEVEL?
000602  001004                          BNE     111$            ; IF NE, NO
000604                                  RETURN                  ; (C-BIT CLEAR)

                                ;
                                ; ERROR CONDITIONS
                                ;
000606                          101$:   EMSG$R  82              ; DUPLICATE OPERATOR
000614                          111$:   EMSG$R  83              ; PRIMARY TEMPLATE OPERATOR

                                        .DSABL  LSB
```

NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES							
ALOCB	= ******	GX	12-488							
BASE	000170	R	#6-125	*11-452	11-453	12-501	12-506	14-633		
DBLK1	000126	R	#6-97	10-376	10-388	10-391				
DBLK2	000140	R	6-97	#6-100	12-502					
DCB	000150	R	#6-102	8-256	*10-381	13-541				
DEACB	= ******	GX	17-745							
DEVHD	= ******	GX	9-321	13-593						
DNAME	000004	R	#6-81	*10-371	*10-372	10-373				
DSKBUF	000000	R	#7-213	11-445						
DUMMY	000152	R	#6-107	8-278	*10-380					
DVALL	000636	R	8-251	#12-485						
DVDEA	001416	R	7-169	7-170	7-171	7-172	7-173	7-174	7-175	#17-742
DVERR	001324	R	7-156	7-157	7-158	7-159	7-160	7-161	7-162	7-163
			7-164	7-165	7-166	7-167	7-168	7-169	7-170	7-171
			7-172	7-173	7-174	7-175	7-176	7-177	#16-699	
DVLBL	000476	R	8-249	#11-436						
DVSVM	000236	R	8-245	#10-368						
DVXFR	000754	R	8-253	#13-539						
D.DSP	= ******	GX	*8-257	9-330						
D.NAM	= ******	GX	9-324	*13-542						
D.PCB	= ******	GX	*8-259							
D.UCB	= ******	GX	8-261	13-552	*13-558					
D.UCBL	= ******	GX	13-580							
D.UNIT	= ******	GX	9-332	9-335	13-544	13-546				
ERRBUF	000017	R	#6-89	16-701						
ERRMSG	000007	R	#6-87	16-713						
ETAB	000156	R	#6-112	10-416						
FIND	000122	R	8-243	#9-319						
FMT4	001404	R	7-157	7-158	7-159	7-160	#7-200			
FMT4A	001434	R	7-156	7-167	7-168	7-177	#7-201			
FMT6	001475	R	7-161	7-162	7-163	7-164	7-166	#7-202		
FMT9	001516	R	7-165	7-169	7-170	7-171	7-172	7-173	7-174	7-175
			7-176	#7-203						
FM.4	= 000000		#7-157	#7-158	#7-159	#7-160				
FM.4A	= 000000		#7-156	#7-167	#7-168	#7-177				
FM.6	= 000000		#7-161	#7-162	#7-163	#7-164	#7-166			
FM.9	= 000000		#7-165	#7-169	#7-170	#7-171	#7-172	#7-173	#7-174	#7-175
FNDEA	001412	R	7-166	#17-741						
FOUND	000166	R	#6-120	8-247	*9-328	10-381				
LENGTH	000136	R	#6-99	*10-395	10-406	*11-453	12-486			
L$$BFLG	= ******	GX	11-448							
L$$BLDZ	= ******	GX	11-450							
L$$BMXZ	= ******	GX	11-450							
L$$BSA	= ******	GX	11-452							
M$$MGE	= 000000		15-661							
REP4A	001446	R	7-156	7-167	7-168	#18-764				
REP4A1	001456	R	7-177	#18-767						
REP4A2	001516	R	18-766	#18-779						
REPG	001600	R	7-165	7-169	7-170	7-171	7-172	7-173	7-174	7-175
			7-176	#19-807	19-809					
RY4D	001270	R	13-556	13-564	#14-633					
S$$MPL	= ******		8-265	13-569	13-585	13-602				

NTLGET -
Symbol t

ALOCB =
ANSWER
A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
C$$CKP=
C$$ORE=
C$$RSH=
D$$BUG=
D$$ISK=
D$$L11=
D$$YNC=
D$$YNM=
E$$xPR=
FL.ECH=
FL.TOP=
FL.WRN=
FMT0
FM.0 =
F$$LVL=
G$$TPP=
G$$TSS=
G$$TTK=
G$$WRD=
I$$RAR=
I$$RDN=
K$$CNT=
K$$CSR=

. ABS.

DATA
Errors d

*** Asse

Work fi
Work fi
Size of
Size of
Operatir

Elapsed
SY:NTLGE

NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES							
S$$MPL	= ******		8-265	13-569	13-585	13-602				
TCB	000144	R	#6-104	8-258	*10-383	13-543				
TEMP	000000	R	#6-79	8-241	8-242	8-244	8-246	8-247	8-248	8-250
			8-252	8-255	8-256	8-258	8-260	8-262	8-264	8-266
			8-267	8-268	8-269	8-270	8-271	8-272	8-273	8-274
			8-275	8-276	8-277	8-278	8-279	8-280	8-281	8-282
			8-283	8-284	8-285	8-286	8-287	8-288	8-289

NTLGET - GET SEND/RECEIVE DATA MACRO V05.03b Saturday 29-Jun-85 00:31 Page 9-1
Symbol table

```
ALOCB = ****** GX        K$$LDC= 000000          LS.FDX= 004000          N$$MCP= 000001          V$$LTR= 001000
ANSWER  000010R     002  K$$TPS= 000074          LS.HDX= 010000          N$$MLL= 000001          X$$DBT= 000000
A$$CHK= 000000           LD$LP = 000000          LS.LIN= 000003          N$$MOV= 000010          $CFUIC= ****** GX
A$$CPS= 000000           LO.CON  000006          LS.LMC= 000007          N$$NCT= 000001          $CRAVL= ****** GX
A$$PRI= 000000           LO.CSR  000006          LS.NTI= 004000          N$$PEM= 000001          $CXUIC= ****** GX
A$$TRP= 000000           LO.CTL  000014          LS.ON = 000011          PUTAST  000320R         $DFUIC= ****** GX
C$$CKP= 000000           LO.LIN  000004          LS.OPT= 000400          P$$P45= 000000          $ERR40  000064R     002
C$$ORE= 000400           LO.PAR  000010          LS.PRO= 000002          P$$WRD= 000000          $FLAGS= ****** GX
C$$RSH= 177564           LO.PRI  000010          LS.PWF= 040000          QASTT = ****** GX       $FUNC = ****** GX
D$$BUG= 177514           LO.SCR  000014          LS.TOP= 002000          Q$$OPT= 000010          $HEADR= ****** GX
D$$ISK= 000000           LO.STA  000015          LS.UNF= 020000          RCVDAT  000000R     002 $LDUIC= ****** GX
D$$L11= 000000           LO.VCT  000012          LS.X25= 000012          RSX$OK= 000000          $NLGET  000146RG
D$$VNC= 00000            LR.AST  000026          LS.11D= 004000          RTSPC   000234R         $NLINI  000000RG
D$$YNM= 000000           LR.CIR  000002          LX.CEX= 000004          R$$DER= 000000          $NLPUT  000236RG
E$$XPR= 000000           LR.CTL  000004          LX.LIN= 000006          R$$K11= 000001          $RFRR = ****** GX
FL.ECH= ****** GX        LR.EFN  000004          LX.PRO= 000005          R$$SND= 000000          $TTO  = ****** GX
FL.TOP= ****** GX        LR.LIN  000002          L$$ASG= 000000          R$$11M= 000000          $XBUF2= ****** GX
FL.WRN= ****** GX        LR.PRM  000030          L$$DRV= 000000          R.VXBA= 000006          $XLEN2= ****** GX
FMTO    000136R     002  LR.PRO  000002          L$$P11= 000001          R.VXTN= 000002          $XXBUF= ****** GX
FM.0  = 000000           LR.STS  000000          L$$11R= 000000          SRSTD = ****** GX       $XXINI= ****** GX
F$$LVL= 000001           LR.TCB  000002          M$$CRB= 000124          S$$WRG= 000000          $XXLEN= ****** GX
G$$TPP= 000000           LR.UNT  000005          M$$CRX= 000000          S$$YSZ= 007600          $$AST = ****** GX
G$$TSS= 000000           LS.AST= 100000          M$$FCS= 000000          S.DABA= 000006          $$LRB   000022RG    002
G$$TTK= 000000           LS.CEX= 000001          M$$MGE= 000000          S.DAEF= 000010          $$PRM = ****** GX
G$$WRD= 000000           LS.CIR= 000010          M$$NET= 000000          S.DATN= 000002          $$TASK= ****** GX
I$$RAR= 000000           LS.CXO= 010000          M$$OVR= 0000[illegible]  T$$KMG= 000000         $$$OST= 000012
I$$RDN= 000000           LS.CX5= 000400          N$$ACC= 00000[illegible]  T$$MIN= 000000         $$$T1 = 000006
K$$CNT= 177546           LS.DLM= 004000          N$$BUF= 000001          U.SCB = ****** GX       .TT0  = ****** GX
K$$CSR= 177546           LS.ECH= 001000          N$$LDV= 000001
```

```
. ABS.  000032    000   (RW,I,GBL,ABS,OVR)
        000432    001   (RW,I,LCL,REL,CON)
DATA    000140    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statist .s


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12800 Words  ( 50 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  syscem: RSX-11M/PLUS

Elapsed time: 00:00:13.09
SY:NTLGET.V2,[132,134]NTLGET/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLGET
```


NTLGET   CREATED BY MACRO ON 29-JUN-85 AT 00:31   PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

```
                                        .SBTTL  KMC LOADER (POWER FAIL)


                                ;+
                                ; $KMCPF - RELOAD ALL MICRO-CODE AFTER A POWER FAILURE
                                ;
                                ; INPUTS:
                                ;       $NTLPT = ADDRESS OF NTL CONTROL DATA BLOCK
                                ;       .CXKMC IN DATA BLOCK CONTAINS ADDRESS OF FIRST "LOAD" LIST
                                ;         BLOCK (ZERO IF NONE ALLOCATED)
                                ;       THE KMC "LOAD" LIST IS PROPERLY UPDATED
                                ;       NO REGISTER SAVING IS NECESSARY
                                ;       FOR "LOAD" LIST FORMAT, SEE THE KMCLL ROUTINE
                                ;
                                ; OUTPUTS:
                                ;       THE "LOAD" LIST IS STEPPED THROUGH, AND, FOR EACH NON-ZERO
                                ;         ENTRY, THE CORRESPONDING KMC IS LOADED FROM ITS MICRO-CODE
                                ;         FILE.
                                ;       C-BIT=SUCCESS/FAILURE.
                                ;-
000036                          $KMCPF::.IF     DF,M$$MGE               ;*-THE FOLLOWING IS MAPPED CODE-*

000036  005067  000012'                 CLR     KMCMP1                  ;M-INDICATE JUST STARTING SCAN

000042                          5$:     .IF     DF,R$$11M
000042                                  SWSTK$  35$                     ;M-SWITCH STACKS, RETURN TO 35$
                                        .ENDC

000046  016700  000012'                 MOV     KMCMP1,R0               ;M-RECOVER MAPPING OF CURRENT LIST
000052  001405                          BEQ     10$                     ;M-BR IF JUST STARTING
000054                                  CALL    $MAPX                   ;M-MAP TO LIST
000060  016703  000014'                 MOV     KMCPN1,R3               ;M-POINT TO ENTRY JUST USED
000064  000427                          BR      23$                     ;M-AND SKIP TO NEXT ENTRY

000066  016700  000000G         10$:    MOV     $NTLPT,R0               ;M-GET MAPPING OF NTL DATA BLOCK
000072                                  CALL    $MAPX                   ;M-MAP TO IT
000076  010203                          MOV     R2,R3                   ;M-POINT AT BEGINNING OF BLOCK
000100  016367  000000G 000012'         MOV     .CXKMC(R3),KMCMP1       ;M-SAVE MAPPING OF FIRST LIST
000106  001435                          BEQ     30$                     ;M-BR IF NO LISTS
000110  016311  000000G                 MOV     .CXKMC(R3),(R1)         ;M-MAP TO FIRST LIST
000114  000404                          BR      18$                     ;M-PROCESS LIST

000116  011367  000012'         15$:    MOV     (R3),KMCMP1             ;M-SAVE MAPPING OF NEXT LIST
000122  001427                          BEQ     30$                     ;M-BR IF NO NEXT LIST
000124  011311                          MOV     (R3),(R1)               ;M-MAP TO NEXT LIST

000126  005723                  18$:    TST     (R3)+                   ;M-POINT AT FIRST ENTRY IN LIST
000130  012767  000007  000006'         MOV     #7,KMCDEV               ;M-LIST HOLDS 7 SLOTS

000136  011367  000000'         20$:    MOV     (R3),KMCNAM             ;M-SAVE THE DEVICE NAME
000142  001007                          BNE     25$                     ;M-BR IF SLOT IS NOT EMPTY

000144  062703  000010          23$:    ADD     #10,R3                  ;M-POINT AT NEXT ENTRY (4 WORDS/ENTRY)
000150  005367  000006'                 DEC     KMCDEV                  ;M-COUNT SLOTS
000154  003370                          BGT     20$                     ;M-GO CHECK THIS SLOT
000156  010203                          MOV     R2,R3                   ;M-POINT AT BEGINNING AGAIN
000160  000756                          BR      15$                     ;M-GO GET NEXT BLOCK IN LIST
```

```
                                        .SBTTL  RAM ERROR PREFIX SUB-FORMATTER

                                ;+
                                ; -KMCRL1- REPLACE DEVICE NAME, CSR, AND AGRGUMENT VALUE WHEN FORMATTING
                                ;
                                ; THE KMCRL1 ROUTINE WILL REPLACE THE FIRST ASTERISK IN THE PREFIX
                                ; STRING WITH THE DEVICE NAME, THE SECOND WITH THE KMC CSR ADDRESS,
                                ; AND THE THIRD WITH THE VALUE OF ARGUMENT 1.  THERE MUST BE
                                ; AT LEAST 3 ASTERISKS IN THE PREFIX STRING.  IT WILL NOT LOOK AT
                                ; THE SUFFIX STRING.
                                ;
                                ; INPUTS:
                                ;       R3 = ADDRESS OF MESSAGE PREFIX STRING
                                ;       R4 = ADDRESS OF MESSAGE BUFFER
                                ;       KMCDEV CONTAINS THE ASCII DEVICE NAME, ENDED BY NULL
                                ;       KMCCSR CONTAINS THE KMC CSR ADDRESS
                                ;       KMCER1 CONTAINS THE THIRD OCTAL ARGUMENT
                                ;
                                ; OUTPUTS:
                                ;       R0, R1, AND R2 ARE DESTROYED
                                ;       R3 = ADDRESS AFTER END OF PREFIX STRING (AFTER NULL)
                                ;       R4 = ADDRESS AT END OF MESSAGE (AT NULL)
                                ;-

                                        .ENABL  LSB

001372  010400                 KMCRL1: MOV     R4,R0           ;PLACE BUFFER POINTER IN R0
001374  012702  000001                 MOV     #1,R2           ;FOR CALL TO $CBOMG (NO ZERO SUPPRESS)
001400                                 CALL    50$             ;MOVE FROM PREFIX TO BUFF, NEXT PAGE
001404  012701  000006'                MOV     #KMCDEV,R1      ;GET ADDRESS OF DEVICE NAME STRING
001410  112120                 10$:    MOVB    (R1)+,(R0)+     ;MOVE IN DEVICE NAME
001412  001376                         BNE     10$             ;BR UNTIL DONE
001414  005300                         DEC     R0              ;BACK UP TO NULL
001416                                 CALL    50$             ;MOVE FROM PREFIX TO BUFF, NEXT PAGE
001422  016701  000002'                MOV     KMCCSR,R1       ;GET THE CSR ADDRESS VALUE
001426                                 CALL    $CBOMG          ;PLACE ASCII FORM IN BUFFER
001432                                 CALL    50$             ;MOVE FROM PREFIX TO BUFF, NEXT PAGE
001436  016701  000016'                MOV     KMCER1,R1       ;GET THE RAM ADDRESS VALUE
001442  000416                         BR      40$             ;FINISH UP, NEXT PAGE
```

```
                                PBLK:   .WORD   VECSYM,0
                                        .IFF
000020  000032' 000000          PBLK:   .WORD   $PRINC,0
                                        .ENDC

000024                          PSYM:   .BLKW   2
000030                          PVAL:   .BLKW   1


                                ; START OF PROCESS VECTOR TABLE

                                        .IF DF  R$$MPL

                                VECSYM: .WORD   VLNSYM,0
                                VTSYM:  .BLKW   2
                                VECVAL: .WORD   0

                                ; VECTOR TABLE SIZE (WORDS)

                                VLNSYM: .WORD   $PRINC,0
                                VLSYM:  .BLKW   2
                                VLNVAL: .WORD   0
                                        .ENDC


                                ;
                                ; PROCESS LINE TABLE SIZE
                                ;
000032  000044' 000000          $PRINC::.WORD   $PREXT,0
000036  126404  046166                  .RAD50  "$$LLIN"
000042                          $PRLLN::.BLKW   1
                                ;
                                ; PROCESS CONTROLLER TABLE SIZE
                                ;
000044  000056' 000000          $PREXT::.WORD   DTBLK,0
000050  126404  012404                  .RAD50  "$$LCNT"
000054                          $PRCLN::.BLKW   1


                                ;
                                ; DCB DISPATCH TABLE ADDRESS
                                ;
000056  000070' 000000          DTBLK:  .WORD   SQBLK,0
000062  126230  076534          DTSYM:  .RAD50  "$XXTBL"
000066                          DTVAL:  .BLKW   1


                                ;
                                ; FORK AND TIMER BLOCKS INITIALIZATION INFORMATION (FOR AUX)
                                ;
000070  000102' 000000          SQBLK:  .WORD   MILBLK,0
000074  125711  074646                  .RAD50  "$SQSRV"
000100                          $SQSRV: .BLKW   1

000102  000114' 000000          MILBLK: .WORD   SECBLK,0
000106  063357  113075                  .RAD50  "PR100M"
000112                          PR100M: .BLKW   1

000114  000126' 000000          SECBLK: .WORD   XBLK1,0
000120  063357  073470                  .RAD50  "PR1SC"
```

$ALECL - ALLOCATE ECL NODE COUNTERS

```
001436                                  BIAS    R1                      ;; SET CORRECT APR BIAS FOR ECL
001446  010165  000044                  MOV     R1,N$ENC+2(R5)          ;; SET END OF LIST POINTER
001452                          40$:    RETURN
                                ;
001454                          101$:   EMSG$R  70                      ; ELSE ALLOCATION FAILURE
        000001                          .END
```


Symbol table

```
                                        .SBTTL  VECTORED EXEC TABLE
000000                                  .PSECT  $$$VEC,D


                        ;+
                        ; -- VECTORED EXEC TABLE
                        ;
                        ; RSX-11M+ SYSTEMS WILL RESOLVE THESE SYMBOLS WITH THE GIN DIRECTIVE,
                        ; RSX-11M SYSTEMS WILL DEFINE THESE VALUES AT TASKBUILD TIME .
                        ;-

                        ; VECTORED RSX EXEC SYMBOLS

000000                  $VECBG::
000000  000000                  .WORD   0               ; VECTORED TABLE INIT FLAG
000002  000000G         SYSNM:: $SYSNM
000004  000000G         TKPS::  $TKPS
000006  000000G         KSAR5:: KISAR5
000010  000000G         KSAR6:: KISAR6
000012  000000G         KSAR7:: KISAR7
000014  000000G         DEUMR:: $DEUMR
000016  000000C         EXRQN:: $EXRQN
                                .IF DF  R$$MPL
                        NCPU::  $NCPU
                        URMTB:: $URMTB
                        K6TAB:: $K6TAB
                        ICAVL:: $ICAVL
                        UMRPT:: $UMRPT
                        ASUMR:: $ASUMR
                        CTLST:: $CTLST
                        HFMSK:: $HFMSK
                        M100Q:: $M100Q
                        CPMSK:: $CPMSK
                        CEX::   $CEX
                        URMST:: $URMST
                        MPPRO:: $MPPRO
                                .ENDC

000020  000000G         FMASK:: $FMASK
000022  000000G         CLKHD:: $CLKHD
000024  000000G         EXSIZ:: $EXSIZ
000026  000000G         CRAVL:: $CRAVL
000030  000000G         TTNS::  $TTNS
000032  000000G         SRSTD:: $SRSTD
000034  000000G         QASTT:: $QASTT
000036  000000G         DEVHD:: $DEVHD
000040  000000G         DEACB:: $DEACB
000042  000000G         USAR7:: UISAR7
000044  000000G         USDR7:: UISDR7
000046  000000G         NONSI:: $NONSI
000050  000000G         DEAC1:: $DEAC1
000052  000000G         LDPWF:: $LDPWF
000054  000000G         PARHD:: $PARHD
000056  000000G         PLGTH:: P.LGTH
000060  000000G         USAR0:: UISAR0
000062  000000G         KSAR0:: KISAR0
```

```
706 000446  004767  000000G         35$:    CALL    $TMP            ; SETUP DDM LINE TABLE
707 000452  103406                          BCS     101$            ; IF CS, FAILURE
708 000454  005767  001266'                 TST     $$HTIM          ; SET UP HELLO AND LISTEN TIMERS?
709 000460  001402                          BEQ     50$             ; BR IF NO
710 000462  004767  000000G                 CALL    $ALTLC          ; SAVE HELLO AND LISTEN TIMERS
711 000466  000207                  50$:    RETURN
712
713                                         ;
714                                         ; ERROR CONDITION
715                                         ;
716 000470  000167  000000G         101$:   JMP     $XLIN           ; DO A PARTIAL "CLEAR LINE"
```


$SDBNM
$SDBSZ
$SLIN
$SLN
$SLTA
$SLTMA
$SLTNM
$SMTSZ
$SPRO
$SRSTD
$STMFC
$SYSNM
$TKPS
$TKTCB
$TK100
$TMP
$TRIB
$TTNS
$T1SCL
$T100C
$T100Q
$URM
$VECBG
$VECLN
$XAVL
$XBIAS
$XBUF2
$XFER
$XLEN2
$XLIN
$XPRO
$X2HLD
$X2MBS
$X2MRT
$X2MWS
$X2RET
$X3CAL
$X3CLR
$X3DBS
$X3DWS
$X3MBS
$X3MCL
$X3MRE
$X3MRS
$X3MWS
$X3RES
$X3RST
$ZTIME
$$APR
$$AST
$$CSR
$$DCHA

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 7

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$SDBNM	= ****** GX	6-208
$SDBSZ	= ****** GX	6-209
$SLIN	000314 R	6-406 #11-669
$SLN	000144 RG	#6-319
$SLTA	000136 RG	#6-316 14-825
$SLTMA	= ****** GX	6-166
$SLTNM	= ****** GX	6-167
$SMTSZ	= 000464 G	#6-340
$SPRO	000070 R	6-405 #8-505
$SRSTD	= ****** GX	5-113
$STMFC	= ****** GX	6-201
$SYSNM	= ****** GX	5-85
$TKPS	= ****** GX	5-86
$TKTCB	= ****** GX	5-136
$TK100	= ****** GX	5-156
$TMP	= ****** GX	8-523 11-686 11-706 14-828
$TRIB	001566 RG	#6-432
$TTNS	= ****** GX	5-112
$T1SCL	= ****** GX	6-186
$T100C	= ****** GX	6-185
$T100Q	= ****** GX	6-184
$URM	000022 RG	#6-269
$VECBG	000000 RG	#5-83 6-215
$VECLN	= 000136 G	#6-215
$XAVL	= ****** GX	6-202 6-204
$XBIAS	= ****** GX	6-203
$XBUF2	001256 RG	#6-416
$XFER	= ****** GX	8-509 11-677 11-694 14-821
$XLEN2	001260 RG	#6-417
$XLIN	= ****** GX	11-716 14-834
$XPRO	= ****** GX	8-564
$X2HLD	001212 RG	#6-339
$X2MBS	001132 RG	#6-336
$X2MRT	001152 RG	#6-337
$X2MWS	001112 RG	#6-335
$X2RET	001172 RG	#6-338
$X3CAL	000120 RG	#6-306
$X3CLR	000122 RG	#6-307
$X3DBS	000110 RG	#6-302
$X3DWS	000114 RG	#6-304
$X3MBS	000112 RG	#6-303
$X3MCL	000130 RG	#6-310
$X3MRE	000132 RG	#6-311
$X3MRS	000134 RG	#6-312
$X3MWS	000116 RG	#6-305
$X3RES	000124 RG	#6-308
$X3RST	000126 RG	#6-309
$ZTIME	= ****** GX	5-126
$$APR	000316 RG	#6-326
$$AST	000002 RG	#6-224
$$CSR	000226 RG	#6-321
$$DCHA	000442 RG	#6-330 6-356

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 8

```
                                        .SBTTL  SCEX - CHECK SET SYS REQUEST BLOCK

                                ;+
                                ; SCEX - CHECK "SET CEX" REQUEST BLOCK
                                ;
                                ; INPUTS:
                                ;       R3=REQUEST BLOCK POINTER
                                ;               .BLKW   1       ; LR.BLK - NETLDR DATA BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       DATA BLOCK ADDRESS STORED IN $NTLPT
                                ;       FL.CEX SET IN $FLAGS
                                ;-
000070  052767  000000G 000000G SCEX:   BIS     #FL.CEX,$FLAGS  ; SET NETLDR MODE

000076  032777  000000G 000000G        BIT     #FE.CEX,@FMASK  ; IS COMM EXEC ALREADY LOADED?

                                        .IF DF  R$$MPL
                                        MOV     FMASK,R0        ; GET $FMASK ADDRESS
                                        ADD     #6,R0           ; $FMASK+6
                                        BIT     #F4.CXD,(R0)    ; HAS COMM EXEC BEEN DEALLOCATED?
                                        BEQ     1$              ; BR IF NO
                                        EMSG$R  85              ; ELSE CAN'T LOAD COMM EXEC
                                1$:
                                        .ENDC

000104  001025                          BNE     10$             ; IF NE, YES
000106  012700  000000G                 MOV     #.HOME,R0       ; ZERO THE NTINIT PARAMETERS
000112  012701  000140                  MOV     #<48.*2>,R1     ; WHICH ARE SITTING IN $$DC4A+4
000116  005020                  2$:     CLR     (R0)+           ; FOR A LENGTH OF 48. DOUBLEWORDS
000120                                  SOB     R1,2$           ;   ...
000124  032767  004000  000004G        BIT     #LS.NTI,$$LRB+4 ; PERFORM NTINIT SETUP ALSO ?
000132  001404                          BEQ     5$              ; IF EQ, NO
000134  052767  000000G 000000G        BIS     #FL.NTI,$FLAGS  ; UPDATE GLOBAL STATUS
000142  000407                          BR      20$             ; NTL FINDS HOME BLOCK IN THIS CASE
000144  011367  000000G          5$:    MOV     (R3),.HOME      ; SAVE DATA BLOCK ADDRESS
000150  001004                          BNE     20$             ; IF NE, NO PROBLEM
000152                                  EMSG$R  91              ; NO DATA BLOCK ADDRESS (NTINIT PROBLEM)
000160  005004                  10$:    CLR     R4              ; NO WORK TO DO!
000162                          20$:    RETURN
```

```
747 001710  126204  000001G                 CMPB    D.UNIT+1(R2),R4 ;* ALL UNITS CHECKED?
748 001714  103346                          BHIS    60$             ;* IF HIS, NO
749 001716  005062  000000G                 CLR     D.DSP(R2)       ;* MARK DEVICE "UNLOADED"
750 001722  005062  000000G                 CLR     D.PCB(R2)       ;*
751 001726                          80$:    RETURN                  ;* BACK TO USER MODE
752 001730  005701                  90$:    TST     R1              ; WERE ANY UNITS BUSY?
753 001732  001403                          BEQ     95$             ; IF EQ, NO
754 001734                                  EMSG$R  Z7              ; DEVICE BUSY
755 001742  010001                  95$:    MOV     R0,R1           ; ANY UNITS MOUNTED?
756 001744  001442                          BEQ     150$            ; IF EQ, NO
757 001746                                  EMSG$R  Z4              ; DEVICE STILL MOUNTED
758
759
760 001754  012701  000003          100$:   MOV     #L.DLC,R1       ; ASSUME DLC
761 001760  032767  000000G 000000G         BIT     #FL.DLC,$FLAGS  ; IS IT A DLC?
762 001766  001002                          BNE     110$            ; IF NE, YES
763 001770  062701  177777                  ADD     #L.DDM-L.DLC,R1 ; IT'S A DDM
764 001774  017702  000000G          110$:  MOV     @SLTMA,R2       ; GET SLT MAPPING TABLE ADDRESS
765
766 002000  017703  000000G                 MOV     @SLTNM,R3       ; AND NUMBER OF SYSTEM LINES
767 002004  001421                          BEQ     140$            ; IF ZERO, NONE TO CHECK
768 002006  012246                  120$:   MOV     (R2)+,-(SP)     ; GET FLAGS WORD ADDRESS ON STACK
769 002010  062716  000000                  ADD     #L.FLG,(SP)     ;
770 002014  032736  040000                  BIT     #LF.RDY,@(SP)+  ; IS THIS LINE STILL READY ?
771 002020  001411                          BEQ     130$            ; IF EQ, NO
772 002022  016246  177776                  MOV     -2(R2),-(SP)    ; GET SLT ADDRESS ON STACK
773 002026  060116                          ADD     R1,(SP)         ; ADD INDEX POINTER
774 002030  123667  000126'                 CMPB    @(SP)+,PDVX     ; DOES THIS LINE REFERENCE OUR PROCESS?
775 002034  001003                          BNE     130$            ; IF NE, NO
776 002036                                  EMSG$   Z5              ; LINE STILL READY (SETS R0 NON-ZERO)
777 002044                          130$:   SOB     R3,120$         ; LOOP <SYSTEM LINE COUNT> TIMES
778 002050  005400                  140$:   NEG     R0              ; SET C-BIT IF ERROR OCCURRED
779 002052                          150$:   RETURN
```

```
                                        .SBTTL  REP7A - REPLACE REQUESTING TASK'S NAME AND CHANNEL NUMBER

                                ;+
                                ;
                                ; REP7A - REPLACE REQUESTING TASK'S NAME AND CHANNEL NUMBER
                                ;
                                ; INPUTS:
                                ;       $$TASK=REQUESTING TASK'S NAME
                                ;       R1=MAPPING TABLE (WITHIN PDV) ADDRESS
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       R3,R4=UPDATED
                                ;
                                ;-

003422  010146                  REP7A:  MOV     R1,-(SP)        ; SAVE CHANNEL TABLE POINTER
003424                                  CALL    REP7            ; REPLACE REQUESTING TASK'S NAME
003430  112314                  10$:    MOVB    (R3)+,(R4)      ; LOOK FOR THE ASTERISK
003432  122724  000052                  CMPB    #'*,(R4)+       ;       ...
003436  001374                          BNE     10$             ;       ...
003440  010400                          MOV     R4,R0           ; COPY BUFFER ADDRESS
003442  112740  000060                  MOVB    #'0,-(R0)       ; SKIP THE ASTERISK
003446  016701  000000G                 MOV     $PDVA,R1        ; GET PDV ADDRESS
003452  062701  000002G                 ADD     #Z.MAP+2,R1     ; POINT AT MAPPING TABLE
003456  162601                          SUB     (SP)+,R1        ; LESS INPUT VALUE
003460  001423                          BEQ     20$             ; IF EQ, NO WORK TO DO
003462  005401                          NEG     R1              ; CONVERT BACK TO A POSITIVE NUMBER
003464  006201                          ASR     R1              ; AND TO A BYTE INDEX
003466  005002                          CLR     R2              ; SUPPRESS LEADING ZEROES
003470                                  CALL    $CBOMG          ; CONVERT TO OCTAL
003474  010004                          MOV     R0,R4           ; SET NEW BUFFER ADDRESS
003476  112314                  15$:    MOVB    (R3)+,(R4)      ; LOOK FOR THE SECOND ASTERISK
003500  122724  000052                  CMPB    #'*,(R4)+       ;       ...
003504  001374                          BNE     15$             ;       ...
003506  010400                          MOV     R4,R0           ; COPY BUFFER ADDRESS
003510  005300                          DEC     R0              ; BACK UP OVER ASTERISK
003512  016701  000000G                 MOV     $PDVA,R1        ; GET PDV ADDRESS
003516  016101  000000G                 MOV     Z.NAM(R1),R1    ; GET PROCESS NAME
003522                                  CALL    $C5TA           ; CONVERT TO ASCII
003526  010004                          MOV     R0,R4           ; SET NEW BUFFER ADDRESS
003530                          20$:    RETURN
```

```
                        .TITLE  NTLSUB - COMMON SUBROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - UTILITY SUBROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

NTLSUB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES								
ALUN$S	#5-59								
CALL	6-175	6-178	7-221	8-260	9-282	9-285	9-293	12-360	12-368
DIR$	#5-59	10-317							
OFF$	#5-87								
QDPB$	#5-87	5-87							
QIOW$	#5-59	5-87							
QIOW$S	#5-59								
RETURN	6-183	7-226	8-264	9-297	10-318	11-338	12-381		
SWSTK$	12-360								

FILEID**NTLSYM

```
                        .TITLE  NTLTMP - TEMPLATE PROCESSING
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE FILE PROCESSING MAIN CONTROL LOOP
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                                ;+
                                ; $TERR - PRINT ERROR MESSAGE
                                ; $TWARN - PRINT WARNING MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=ERROR MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;       R0,R1=DESTROYED
                                ;-

                                        .ENABL  LSB

000622                          $TWARN::
000622  012701  000472'                 MOV     #WPFX,R1        ; SET PREFIX ADDRESS
000626  005767  000000G                 TST     $FLAGS          ; ARE WARNING MESSAGES ENABLED?
000632  100412                          BMI     10$             ; IF MI, YES
000634  000445                          BR      60$             ; ELSE JUST EXIT WITH C-BIT SET
000636                          $TWE::
000636  032767  000000G 000000G         BIT     #FL.LLC,$FLAGS  ; WARNING IF LLC
000644  001366                          BNE     $TWARN          ; FATAL IF DDM OR DLC
000646                          $TERR::
000646  052767  100000  000000'         BIS     #F1.ERR,$FLAG1  ; INDICATE ERROR HAS OCCURRED
000654  012701  000507'                 MOV     #EPFX,R1        ; SET PREFIX ADDRESS
000660                          10$:    CALL    $SAVAL          ; SAVE ALL REGISTERS
000664  010005                          MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
000666  012704  000361'                 MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
000672  112124                  20$:    MOVB    (R1)+,(R4)+     ; COPY THE PREFIX
000674  001376                          BNE     20$             ;  ...
000676  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000700  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
000702                                  CALL    @(R5)+          ; DO THE FORMATTING
000704  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
000706  001403                          BEQ     40$             ; IF EQ, YES
000710  112324                  30$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
000712  001376                          BNE     30$             ;  ...
000714  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000716  012503                  40$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
000720  112524                  50$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
000722  001376                          BNE     50$             ;  ...
000724  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000726  012700  000354'                 MOV     #ERRMSG,R0      ; GET ERROR MESSAGE ADDRESS
000732  010401                          MOV     R4,R1           ; COPY END ADDRESS
000734  160001                          SUB     R0,R1           ; GIVING LENGTH
000736                                  CALL    $TTOUT          ; PRINT IT
000742  005703                          TST     R3              ; SHOULD SECOND LINE BE PRINTED?
000744  001401                          BEQ     60$             ; IF EQ, NO
000746                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
000750  000261                  60$:    SEC                     ; RETURN C-BIT SET
000752                                  RETURN

                                        .DSABL  LSB


000001                                  .END
```


Symbol table

NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
R$$11M	= 000000	6-72								
SLHD	= ****** GX	13-574								
STBEXT	000000 R	#6-76	10-386							
S$$BAS	= ******	7-156	7-156	7-157	7-157	7-158	7-158	7-159	7-159	7-160
		7-160	7-161	7-161	7-162	7-162	7-163	7-163	7-164	7-164
		7-165	7-165	7-166	7-166	7-167	7-167	7-168	7-168	7-169
		7-169	7-170	7-170	7-171	7-171	7-172	7-172	7-173	7-173
		7-174	7-174	7-175	7-175	7-176	7-176	7-177	7-177	
TABEND	000174 R	#6-135	*12-500	14-640						
TS$NHD	= ****** GX	11-448								
UNITS	000172 R	#6-130	8-260	*9-339	*13-550					
U.DCB	= ****** GX	*13-566								
U.RED	= ****** GX	*13-567								
U.SCB	= ****** GX	13-560	*13-568							
VAD	001300 R	13-553	13-561	13-581	#14-635					
$AZFS	= ****** GX	10-387	11-439							
$CAT5	= ****** GX	10-375								
$CLDEQ	= ****** GX	7-165	8-255	17-742	17-748					
$CLIOS	= ****** GX	11-441	11-462	13-610						
$CLOPE	= ****** GX	11-440								
$CUR.N	= ****** GX	7-163	7-166							
$C5TA	= ****** GX	18-774	18-787	18-789						
$DFN2	= ****** GX	10-370								
$ERR2A	000176 R	#7-156	10-396							
$ERR2B	000220 R	6-112	#7-157							
$ERR2C	000256 R	6-113	#7-158							
$ERR2D	000314 R	6-114	#7-159							
$ERR2E	000342 R	6-115	#7-160							
$ERR2F	000370 R	#7-161	11-460							
$ERR2H	000426 R	#7-162	11-461							
$ERR2J	000456 R	#7-163	11-463							
$ERR2K	000530 R	#7-164	11-464							
$ERR2L	000572 R	#7-165	12-513							
$ERR2M	000626 R	#7-166	13-611							
$ERR2N	000664 R	#7-167	10-404							
$ERR2P	000700 R	#7-168	10-408							
$ERR2Q	000736 R	#7-169	9-346	13-612						
$ERR2R	000772 R	#7-170	13-613							
$ERR2S	001044 R	#7-171	13-616							
$ERR2T	001074 R	#7-172	13-617							
$ERR2U	001146 R	#7-173	13-618							
$ERR2V	001220 R	#7-174	13-621							
$ERR2W	001250 R	#7-175	13-622							
$ERR2X	001322 R	#7-176	9-345							
$ERR2Y	001362 R	#7-177	8-279							
$FNDEQ	001436 R	7-162	7-163	7-164	#17-747					
$FNOUT	= ****** GX	7-157	7-158	7-159	7-160	7-161	17-741	17-747		
$IOSB	= ****** GX	11-462	13-610							
$LBUF	= ****** GX	14-634	14-638	17-743						
$LLEN	= ****** GX	*12-497	17-744							
$NAME	= ****** GX	10-368	18-773							
$NLCDD	= ****** GX	8-272								

NTLGET

SYMBOL C

SYMBOL

ALOCB
ANSWER
FL.ECH
FL.TOP
FL.WRN
FMTO
FM.O
I$$AS
LO.CON
LO.CSR
LO.CTL
LO.LIN
LO.PAR
LO.PRI
LO.SCR
LO.STA
LO.VCT
LR.AST
LR.CIR
LR.CTL
LR.EFN
LR.LIN
LR.PRM
LR.PRO
LR.STS
LR.TCB
LR.UNT
LS.AST
LS.CEX
LS.CIR
LS.CXO
LS.CX5
LS.DLM
LS.ECH
LS.FDX
LS.HDX
LS.LIN
LS.LMC
LS.NTI
LS.ON
LS.OPT
LS.PRO
LS.PWF
LS.TOP
LS.UNF
LS.X25
LS.11D
LX.CEX
LX.LIN
LX.PRO
N$$VCT
PUTAST

NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 3

CREF 04.00

NTLGET

NTLGET CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES			
ALOCB	= ****** GX	9-234			
ANSWER	000010 R	#5-86	*8-200	*8-201	8-208
FL.ECH	= ****** GX	6-143	7-179		
FL.TOP	= ****** GX	6-143	7-179		
FL.WRN	= ****** GX	6-143	7-179		
FMTO	000136 R	5-99	#5-109		
FM.O	= 000000	#5-99			
I$$AS	= ******	5-62	7-172	8-194	9-225
LO.CON	000006	#5-60			
LO.CSR	000006	#5-60			
LO.CTL	000014	#5-60			
LO.LIN	000004	#5-60			
LO.PAR	000010	#5-60			
LO.PRI	000010	#5-60			
LO.SCR	000014	#5-60			
LO.STA	000015	#5-60			
LO.VCT	000012	#5-60			
LR.AST	000026	#5-60	7-176		
LR.CIR	000002	#5-60			
LR.CTL	000004	#5-60			
LR.EFN	000004	#5-60			
LR.LIN	000002	#5-60			
LR.PRM	000030	#5-60	7-177		
LR.PRO	000002	#5-60			
LR.STS	000000	#5-60			
LR.TCB	000002	#5-60			
LR.UNT	000005	#5-60			
LS.AST	= 100000	#5-60	7-174		
LS.CEX	= 000001	#5-60			
LS.CIR	= 000010	#5-60			
LS.CXO	= 010000	#5-60			
LS.CX5	= 000400	#5-60			
LS.DLM	= 004000	#5-60			
LS.ECH	= 001000	#5-60			
LS.FDX	= 004000	#5-60			
LS.HDX	= 010000	#5-60			
LS.LIN	= 000003	#5-60			
LS.LMC	= 000007	#5-60			
LS.NTI	= 004000	#5-60			
LS.ON	= 000011	#5-60			
LS.OPT	= 000400	#5-60			
LS.PRO	= 000002	#5-60			
LS.PWF	= 040000	#5-60			
LS.TOP	= 002000	#5-60			
LS.UNF	= 020000	#5-60			
LS.X25	= 000012	#5-60			
LS.11D	= 004000	#5-60			
LX.CEX	= 000004	#5-60			
LX.LIN	= 000006	#5-60			
LX.PRO	= 000005	#5-60			
N$$VCT	= ******	9-236			
PUTAST	000320 R	8-197	#9-227		

NTLKMC
KMC LOA

309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365

NTLGET CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

NTLKMC
KMC LOA

```
                                        25$:    MOV     2(R3),KMCCSR            ;M-SAVE THE DEVICE CSR ADDRESS
                                                MOVB    3(R3),KMCNUM            ;M-SAVE THE CONTROLLER NUMBER
                                                MOV     R3,KMCPN1               ;M-SAVE THE POINTER TO THE ENTRY

                                        30$:    .IF     DF,R$$11M
                                                RETURN                          ;M-RETURN TO NORMAL AT 35$
                                                .IFF
                                                CALL    @(SP)+                  ;M-RESTORE MAPPING VIA $MAPX
                                                .ENDC

                                        35$:    TST     KMCMP1                  ;M-DID WE COME TO THE END YET?
                                                BEQ     40$                     ;M-BR IF YES
                                                MOV     KMCNAM,R0               ;M-GET THE DEVICE NAME (RAD50)

                                                .IF DF  R$$MPL

                                                TSTB    $NCPU                   ;M+ IS THIS A MULTI-PROCESSOR SYSTEM ?
                                                BEQ     39$                     ;M+ IF EQ, NO
                                                MOV     $SLTMA,R1               ;M+ GET THE SLT TABLE ADDRESS
                                                MOV     $SLTNM,R3               ;M+ AND THE NUMBER OF TABLE ENTRIES

                                        36$:    MOV     (R1)+,R2                ;M+ GET THE NEXT ENTRY
                                                CMPB    KMCNUM,L.CTL(R2)        ;M+ IS THIS THE RIGHT CONTROLLER ?
                                                BNE     37$                     ;M+ IF NE, NO
                                                MOVB    L.DDM(R2),R4            ;M+ GET THE DDM PDV INDEX
                                                ADD     $PDVTA,R4               ;M+ GET ADDRESS OF PDV ENTRY
                                                MOV     (R4),R4                 ;M+ POINT TO ENTRY
                                                CMP     R0,Z.NAM(R4)            ;M+ IS THIS THE RIGHT ENTRY ?
                                                BEQ     38$                     ;M+ IF EQ, YES
                                        37$:    SOB     R3,36$                  ;M+ LOOP <$SLTNM> TIMES
                                                BR      101$                    ;M+ CAN'T FIND UNIBUS RUN MASK

                                        38$:    MOV      .KRBA(R2),R2           ;M+ GET THE KRB ADDRESS
                                                MOV     K.URM(R2),$URM          ;M+ PICK UP THE UNIBUS RUN MASK
                                        39$:                                    ;M+
                                                .ENDC

                                                MOV     KMCCSR,R4               ;M-GET THE DEVICE CSR
                                                CALL    KMCLD                   ;M-LOAD THE KMC
                                                BR      5$                      ;M-CHECK NEXT LIST ENTRY

                                                .IFF                            ;*-THE FOLLOWING IS UNMAPPED CODE-*

                                                MOV     $NTLPT,R3               ;U-GET POINTER TO NTL DATA BLOCK
                                                ADD     #.CXKMC,R3              ;U-POINT AT LIN' TO FIRST LIST BLOCK

                                        5$:     MOV     (R3),R3                 ;U-GET POINTER TO LIST BLOCK
                                                BEQ     40$                     ;U-BR IF NO MORE BLOCKS
                                                MOV     R3,R1                   ;U-COPY POINTER TO BLOCK
                                                TST     (R1)+                   ;U-SET POINTER TO FIRST ENTRY
                                                MOV     #7,R2                   ;U-THERE ARE 7 ENTRIES PER BLOCK

                                        10$:    MOV     (R1)+,R0                ;U-GET DEVICE NAME
                                                BEQ     20$                     ;U-BR IF EMPTY SLOT
                                                MOV     (R1)+,R4                ;U-GET CSR ADDRESS
```



```
                                                .IF DF  R$$MPL

                                                TSTB    $NCPU                   ;M+ IS THIS A MULTI-PROCESSOR SYSTEM ?
                                                BEQ     39$                     ;M+ IF EQ, NO
                                                MOV     $SLTMA,R1               ;M+ GET THE SLT TABLE ADDRESS
                                                MOV     $SLTNM,R3               ;M+ AND THE NUMBER OF TABLE ENTRIES

                                        36$:    MOV     (R1)+,R2                ;M+ GET THE NEXT ENTRY
                                                CMPB    KMCNUM,L.CTL(R2)        ;M+ IS THIS THE RIGHT CONTROLLER ?
                                                BNE     37$                     ;M+ IF NE, NO
                                                MOVB    L.DDM(R2),R4            ;M+ GET THE DDM PDV INDEX
                                                ADD     $PDVTA,R4               ;M+ GET ADDRESS OF PDV ENTRY
                                                MOV     (R4),R4                 ;M+ POINT TO ENTRY
                                                CMP     R0,Z.NAM(R4)            ;M+ IS THIS THE RIGHT ENTRY ?
                                                BEQ     38$                     ;M+ IF EQ, YES
                                        37$:    SOB     R3,36$                  ;M+ LOOP <$SLTNM> TIMES
                                                BR      101$                    ;M+ CAN'T FIND UNIBUS RUN MASK

                                        38$:    MOV      .KRBA(R2),R2           ;M+ GET THE KRB ADDRESS
                                                MOV     K.URM(R2),$URM          ;M+ PICK UP THE UNIBUS RUN MASK
                                        39$:                                    ;M+
                                                .ENDC

                                                MOV     KMCCSR,R4               ;M-GET THE DEVICE CSR
                                                CALL    KMCLD                   ;M-LOAD THE KMC
                                                BR      5$                      ;M-CHECK NEXT LIST ENTRY

                                                .IFF                            ;*-THE FOLLOWING IS UNMAPPED CODE-*

                                                MOV     $NTLPT,R3               ;U-GET POINTER TO NTL DATA BLOCK
                                                ADD     #.CXKMC,R3              ;U-POINT AT LIN' TO FIRST LIST BLOCK

                                        5$:     MOV     (R3),R3                 ;U-GET POINTER TO LIST BLOCK
                                                BEQ     40$                     ;U-BR IF NO MORE BLOCKS
                                                MOV     R3,R1                   ;U-COPY POINTER TO BLOCK
                                                TST     (R1)+                   ;U-SET POINTER TO FIRST ENTRY
                                                MOV     #7,R2                   ;U-THERE ARE 7 ENTRIES PER BLOCK

                                        10$:    MOV     (R1)+,R0                ;U-GET DEVICE NAME
                                                BEQ     20$                     ;U-BR IF EMPTY SLOT
                                                MOV     (R1)+,R4                ;U-GET CSR ADDRESS
```

```
                                        .SBTTL  RAM ERROR SUFFIX SUB-FORMATTER
                                ;+
                                ; -KMCRL2- REPLACE WITH ARGUMENTS 2 AND 3 WHEN FORMATTING
                                ;
                                ; THE KMCRL2 ROUTINE WILL REPLACE THE FIRST ASTERISK IN THE SUFFIX
                                ; STRING WITH THE VALUE IN ARGUMENT 2, AND THE SECOND WITH THE VALUE
                                ; IN ARGUMENT 3.  THERE MUST BE AT LEAST TWO ASTERISKS IN THE SUFFIX
                                ; STRING.  IT WILL NOT LOOK AT THE PREFIX STRING.
                                ;
                                ; INPUTS:
                                ;       R4 = ADDRESS OF MESSAGE BUFFER
                                ;       R5 = ADDRESS OF MESSAGE SUFFIX STRING
                                ;       KMCER2 CONTAINS THE VALUE OF ARGUMENT 2
                                ;       KMCER3 CONTAINS THE VALUE OF ARGUMENT 3
                                ;
                                ; OUTPUTS:
                                ;       R0, R1, AND R2 ARE DESTROYED
                                ;       R3 = ADDRESS AFTER END OF SUFFIX STRING (AFTER NULL)
                                ;       R4 = ADDRESS AT END OF MESSAGE (AT NULL)
                                ;-
001444  010503          KMCRL2: MOV     R5,R3                   ;PLACE SUFFIX POINTER IN R3
001446  010400                  MOV     R4,R0                   ;PLACE BUFFER POINTER IN R0
001450  012702  000001          MOV     #1,R2                   ;FOR $CBOMG, NO ZERO SUPPRESS
001454                          CALL    50$                     ;MOVE FROM SUFFIX TO BUFFER
001460  016701  000020'          MOV     KMCER2,R1               ;GET VALUE OF ARGUMENT 2
001464                          CALL    $CBOMG                  ;PLACE IN BUFFER
001470                          CALL    50$                     ;MOVE FROM SUFFIX TO BUFFER
001474  016701  000022'          MOV     KMCER3,R1               ;GET VALUE OF ARGUMENT 3

001500                  40$:    CALL    $CBOMG                  ;PLACE VALUE IN BUFFER
001504  112320          45$:    MOVB    (R3)+,(R0)+             ;FINISH MOVING TO BUFFER
001506  001376                  BNE     45$                     ;UNTIL FINISHED
001510  005300                  DEC     R0                      ;BACK UP TO NULL
001512  010004                  MOV     R0,R4                   ;PLACE VALUE BACK IN R4
001514                          RETURN

                                ;
                                ; MOVE TO BUFFER UNTIL ASTERISK ENCOUNTERED
                                ;
001516  112310          50$:    MOVB    (R3)+,(R0)              ;MOVE IN A BYTE
001520  122720  000052          CMPB    #'*,(R0)+               ;WAS IT AN ASTERISK?
001524  001374                  BNE     50$                     ;BE IF NO
001526  005300                  DEC     R0                      ;BACK UP OVER ASTERISK
001530                          RETURN                          ;FOUND IT

                                        .DSABL  LSB
```

```
    210 000124                          PR1SC:  .BLKW   1
    211
    212                                 ;
    213                                 ; COMM EXEC SYMBOLS USED FOR PROCESS VALIDATION
    214                                 ;
    215 000126  000140' 000000          XBLK1:  .WORD   XBLK2,0
    216 000132  124473  007544                  .RAD50  "$CCBRT"
    217 000136  000000                          .WORD   0
    218 000140  000152' 000000          XBLK2:  .WORD   XBLK3,0
    219 000144  124473  006654                  .RAD50  "$CCBGT"
    220 000150  000000                          .WORD   0
    221 000152  000000  000000          XBLK3:  .WORD   0,0
    222 000156  126344  035115                  .RAD50  "$ZTIME"
    223 000162  000000                          .WORD   0
    224
    225                                 ;
    226                                 ; SYMBOL TABLE ERRORS
    227                                 ;
    228 000164  000000G                 ETAB1:  .WORD   $ERR44          ; OPEN FAILURE
    229 000166  000000G                         .WORD   $ERR45          ; READ FAILURE
    230 000170  000000G                         .WORD   $ERR46          ; READ OVERFLOW
    231 000172  000000G                         .WORD   $ERR47          ; BAD STB FORMAT
    232
    233                                 ;
    234                                 ; PROCESS I/O ERRORS
    235                                 ;
    236 000174  000D00G                 ETAB2:  .WORD   $ERR51          ; OPEN FAILURE
    237 000176  000101                          .WORD   101             ; GET ATTRIBUTES FAILED
    238 000200  000000G                         .WORD   $ERR53          ; FILE NOT CONTIGUOUS
    239 000202  000000G                         .WORD   $ERR54          ; LABEL BLOCK READ FAILURE
    240 000204  000000G                         .WORD   $ERR55          ; INVALID IMAGE (HEADER, OVERLAID, ETC.)
    241 000206  000000G                         .WORD   $ERR56          ; PROCESS NOT BUILD FOR APR5
    242 000210  000000G                         .WORD   $ERR57          ; PROCESS EXCEEDS 4K WORDS
    243 000212  000000G                         .WORD   $ERR58          ; PROCESS EXTENSION EXCEEDS 4K WORDS
    244 000214  000000G                         .WORD   $ERR59          ; PARTITION NOT IN SYSTEM
    245 000216  000000G                         .WORD   $ERR60          ; BASE ADDRESS MISMATCH
    246 000220  000000G                         .WORD   $ERR61          ; PARTITION TOO SMALL
    247 000222  000000G                         .WORD   $ERR62          ; NOT COMMON PARTITION
    248 000224  000000G                         .WORD   $ERR63          ; PARTITION BUSY
    249 000226  000000G                         .WORD   $ERR64          ; CEX PARTITION NOT IN EXEC SPACE
    250 000230  000000G                         .WORD   $ERR65          ; SUB-PCB ALLOCATION FAILURE
    251 000232  000000G                         .WORD   $ERR66          ; MAIN PARTITION TOO FRAGMENTED
    252 000234  000000G                         .WORD   $ERR67          ; PROCESS READ FAILURE
    253
    254 000000                          .PSECT
```



```
AT$ACL
AT$ADN
AT$AUP
AT$CYC
AT$LV1
AT$LV2
AT$NEX
AT$UP
AXSYM
A$CIR
A$LEN
A$NID
A$TM
A$TMI
A$TSZ
A$TYP
A$$CHK
A$$CPS
A$$PRI
A$$TRP
CMFRK
C$LEN
C$NID
C$NXH
C$TIM
C$$CKF
C$$ORE
C$$RSH
DECPT
DSPTAE
DTBLK
DTSYM
DTVAL
D$AMX(
D$AMXH
D$ANN
D$BRPF
D$BRTN
D$DELF
D$DELW
D$END
D$FNB
D$HIOF
D$HOST
D$INA(
D$INC1
D$IPL
D$LID
D$LNAM
D$LNUM
D$LST
D$MAX(
D$MAXH
D$MAX1
D$MLL
D$MNOI
D$NA
```

Symbol table

```
AT$ACL= 000100          D$NBEA  000056          LF.MTP= 000020          NTLPT = ****** GX       PBLK    000020R     002
AT$ADN= 000040          D$NBRA  000054          LF.PAC= 000200          N$ACQ   000000          PRSYM   000462R
AT$AUP= 000020          D$NEND= 000054          LF.RDY= 040000          N$ACTL  000032          PR1SC   000124R     002
AT$CYC= 000004          D$NLN   000030          LF.REA= 010000          N$ADJ1  000072          PR100M  000112R     002
AT$LV1= 000002          D$NN    000060          LF.SER= 000040          N$ADJ2  000074          PSYM    000024R     002
AT$LV2= 000001          D$OUTT  000043          LF.TIM= 000010          N$CACH  000062          PVAL    000030R     002
AT$NEX= 000200          D$RETF  000050          LF.UNL= 020000          N$CIR   000034          P$$P45= 000000
AT$UP = 000010          D$RNN   000002          LF.X2P= 000000          N$CRC   000120          P$$WRD= 000000
AXSYM   000234R         D$RTMR  000076          LN.CLO= 000000          N$DLA   000020          P.BLKS= ****** GX
A$CIR   000003          D$SEG   000036          LN.DUM= 000005          N$DLY   000014          P.REL = ****** GX
A$LEN   000004          D$SER   000032          LN.LOA= 000004          N$ELEN  000054          Q$$OPT= 000010
A$NID   000000          D$SQRL  000052          LN.LOO= 000003          N$ENC   000042          R$$DER= 000000
A$TM    000003          D$$BUG= 177514          LN.OAU= 000003          N$ERRC  000022          R$$K11= 000001
A$TMI   000002          D$$ISK= 000000          LN.OFF= 000001          N$FLG   000005          R$$SND= 000000
A$TSZ   000000          D$$L11= 000001          LN.ON = 000000          N$FNC   000006          R$$11M= 000000
A$TYP   000002          D$$SYNC= 000000         LN.OOP= 000004          N$GENQ  000052          SECBLK  000114R     002
A$$CHK= 000000          D$$SYNM= 000000         LN.OPE= 000001          N$GTM   000015          SF.ACT= 000200
A$$CPS= 000000          ETAB1   000164R     002 LN.REF= 000002          N$HC1   000052          SF.ENA= 000100
A$$PRI= 000000          ETAB2   000174R     002 LN.SER= 000002          N$HC2   000056          SF.LPB= 000004
A$$TRP= 000000          E$NBR   000014          LN.STA= 000017          N$HIGH  000033          SF.MFL= 000040
CMFRK = ****** GX       E$NBS   000020          LN.SUB= 000360          N$LLT   000026          SF.PAC= 000020
C$LEN   000012          E$NCR   000034          LN.TRI= 000006          N$LLTM  000024          SF.REA= 000010
C$NID   000000          E$NCS   000036          L$$ASG= 000000          N$LVC   000036          SF.SER= 000001
C$NXH   000002          E$NIC   000044          L$$DRV= 000000          N$LV1   000002          SF.SVC= 000002
C$TIM   000010          E$NLEN  000050          L$$P11= 000001          N$LV2   000010          SF.UNL= 000040
C$$CKP= 000000          E$NLLA  000012          L$$11R= 000000          N$MBXQ  000050          SQBLK   000070R     002
C$$ORE= 000400          E$NLNK  000000          L.COST  000015          N$MHC1  000036          STBERR  000756R
C$$RSH= 177564          E$NML   000040          L.CTL   000012          N$MHC2  000044          STBEXT  000006R     002
DECPT = ****** GX       E$NMR   000024          L.CVA   177776          N$PLD   000016          S$$WRG= 000000
DSPTAB  000012R     002 E$NMS   000030          L.DDM   000002          N$PLLT  000030          S$$YSZ= 007600
DTBLK   000056R     002 E$NNOD  000002          L.DDS   000004          N$PRI   000076          S.COST  000001
DTSYM   000062R     002 E$NRT   000042          L.DLC   000003          N$ROA1  000022          S.FLG   000000
DTVAL   000066R     002 E$NRTP  000005          L.DLM   000006          N$ROA2  000030          S.LEN   000004
D$AMXC  000072          E$NSEG  000010          L.DLS   000010          N$RTMX  000014          S.NMST  000002
D$AMXH  000074          E$NTIM  000046          L.FLG   000000          N$RTM1  000014          S.OWNR  000003
D$ANN   000000          E$NUSE  000004          L.KRBA  000016          N$RTM2  000015          TEMP    000000R     002
D$BRPR  000102          E$STRT  000006          L.LEN = 000022          N$RT1   000000          TSKERR  000226R
D$BRTM  000100          E$$XPR= 000000          L.MPF   000022          N$RT2   000006          T$FLAG  000044
D$DELF  000045          F$$LVL= 000001          L.NMST  000020          N$SLA   000016          T$LIF   000013
D$DELW  000046          G$$TPP= 000000          L.NSTA  000014          N$SNOD  000012          T$LIFL  000013
D$END = 000104          G$$TSS= 000000          L.OWNR  000021          N$TCTL  000112          T$LIFO  000013
D$FNB   000034          G$$TTK= 000000          L.UNT   000013          N$TIM   000004          T$LIFS  000013
D$HIOR  000024          G$$WRD= 000000          MILBLK  000102R     002 N$TLC   000100          T$LIN   000000
D$HOST  000022          I$$RAR= 000000          M$$CRB= 000124          N$TNC   000114          T$LIPS  000006
D$INAC  000044          I$$RDN= 000000          M$$CRX= 000000          N$TRC   000106          T$LLD   000012
D$INCT  000042          K$$CNT= 177546          M$$FCS= 000000          N$TTCB  000110          T$LLDC  000045
D$IPL   000051          K$$CSR= 177546          M$$MGE= 000000          N$VCB   000010          T$LLDL  000012
D$LID   000020          K$$LDC= 000000          M$$NET= 000000          N$VER   000066          T$LLDO  000012
D$LNAM  000006          K$$TPS= 000074          M$$OVR= 000000          N$XLEN  000124          T$LLDS  000012
D$LNUM  000014          LD$LP = 000000          NF$BLK= 000100          N$$ACC= 000001          T$LLEN  000046
D$LST   000047          LF.ACT= 100000          NF$DMO= 000010          N$$BUF= 000001          T$LOPR  000002
D$MAXC  000064          LF.BRO= 000400          NF$MOU= 000040          N$$LDV= 000001          T$LTCL  000024
D$MAXH  000066          LF.BWT= 000007          NF$RST= 000002          N$$MCP= 000001          T$LTIM  000026
D$MAXV  000070          LF.ENA= 002000          NF$SCN= 000020          N$$MLL= 000001          T$LTPR  000014
D$MLL   000040          LF.LPB= 001000          NF$SHU= 000004          N$$MOV= 000010          T$LTPS  000020
D$MNOD  000041          LF.MDC= 000100          NF$TIM= 000200          N$$NCT= 000001          T$NAPL  000004
D$NA    000062          LF.MFL= 004000          NODCTR  000004R     002 N$$PEM= 000001          T$NFE   000000
```


Symbol table

```
126 000064  000000G                 ZTIME:: $ZTIME
127 000066  000000G                 CMFRK:: $CMFRK
128
129                                         .IF DF R$$MPL
130                                 SKS5::  S.KS5
131                                 SKTB::  S.KTB
132                                 UMRHD:: $UMRHD
133                                         .ENDC
134
135 000070  000000G                 SLHD::  S.LHD
136 000072  000000G                 TKTCB:: $TKTCB
137 000074  000000G                 NTUIC:: $NTUIC
138 000076  000000G                 $MXVCT::V$$CTR
139
140                                 ; VECTORED COMM EXEC SYMBOLS
141
142 000100  000000G                 NBIAS:: $NBIAS
143 000102  000000G                 MAXOV:: $MAXOV
144 000104  000000G                 RDBTH:: $RDBTH
145 000106  000000G                 CEPWR:: $CEPWR
146 000110  000000G                 SDBCT:: $SDBCT
147 000112  000000G                 RDBCT:: $RDBCT
148 000114  000000G                 CCBCT:: $CCBCT
149 000116  000000G                 CCBNM:: $CCBNM
150 000120  000000G                 LTMFC:: $LTMFC
151 000122  000000G                 CCBAF:: $CCBAF
152 000124  000000G                 RDBAF:: $RDBAF
153 000126  000000G                 LDBAF:: $LDBAF
154 000130  000000G                 SDBAF:: $SDBAF
155 000132  000000G                 CCBAL:: $CCBAL
156 000134  000000G                 TK100:: $TK100
157 000136  000000G                 LLCTA:: $LLCTA
158 000140  000000G                 PDVTA:: $PDVTA
159
```

```
                                        .SBTTL  AIC - TEST FOR ALREADY IN CORE

                                ;+
                                ;
                                ; AIC - TEST FOR PROCESS ALREADY IN CORE
                                ;
                                ; INPUTS:
                                ;       R1 - PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       $BIAS - BASE ADDRESS OF PROCESS
                                ;       $GCD (IAS) - GCD ADDRESS
                                ;       $PCB (RSX) - PCB ADDRESS
                                ;
                                ;-

000474  010167  000062'         AIC:    MOV     R1,$PDVA        ; SAVE PDV ADDRESS
000500  116102  000000G                 MOVB    Z.FLG(R1),R2    ; GET PROCESS TYPE FROM PDV STATUS BYTE
000504  042702  177774                  BIC     #^C<FL.DLC!FL.DDM>,R2   ;
000510  042767  000003  001232'         BIC     #<FL.DLC!FL.DDM>,$FLAGS ; UPDATE NETLDR STATUS ACCORDINGLY
000516  050267  001232'                 BIS     R2,$FLAGS       ;
000522  016167  000000G 000064'         MOV     Z.NAM(R1),$NAME ; SET PROCESS NAME (RAD50)

                                        .IF DF  R$$11D ! I$$AS

                                        MOV     Z.GCD(R1),R2    ; GET GCD ADDRESS
                                        BEQ     10$             ; IF EQ, PROCESS IS NOT IN CORE
                                        MOV     G.BA(R2),$BIAS  ; COPY BASE ADDRESS
                                        MOV     R2,$GCD         ; MAKE SURE $GCD HAS THE PROCESS REGION GCD ADDRESS

                                        .IFF

000530  016102  000000G                 MOV     Z.PCB(R1),R2    ; GET PCB ADDRESS
000534  001405                          BEQ     10$             ; IF EQ, PROCESS IS NOT IN CORE
000536  016267  000000G 000066'         MOV     P.REL(R2),$BIAS ; COPY RELOCATION BIAS
000544  010267  000070'                 MOV     R2,$PCB         ; MAKE SURE $PCB HAS THE PROCESS PCB ADDRESS

                                        .ENDC

000550  000207                  10$:    RETURN
```


$$DEV
$$DPR
$$EXT
$$HTIM
$$INC
$$LINX
$$LSAD
$$LTIM
$$MTP
$$MUX
$$NODC
$$PAR
$$PCHA
$$PDIN
$$PRI
$$PRM
$$SCSR
$$SNUM
$$TASK
$$TBL
$$VECT
.ALLOC
.AMAXC
.AMAXH
.BFPTR
.CCBNM
.CCBSZ
.FUDGE
.HOME
.MAXC
.MAXH
.MXPRO
.NA
.NAMBF
.NBEA
.NBRA
.NN
.NOMAP
.NTPSZ
.PCB
.PNAM1
.PNAM2
.PUMR
.RDBNM
.RDBSZ
.RDBTH
.ROUTB
.ROUTM
.RPRI
.SDBNM
.SDBSZ
.UMRBA

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 8

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES	
$$DEV	000100	RG	#6-298	8-541
$$DPR	000376	RG	#6-327	
$$EXT	000030	RG	#6-272	
$$HTIM	001266	RG	#6-430	11-708
$$INC	000072	RG	#6-296	
$$LINX	000104	RG	#6-300	
$$LSAD	001110	RG	#6-334	
$$LTIM	001426	RG	#6-431	
$$MTP	000234	RG	#6-324	
$$MUX	000400	RG	#6-328	
$$NODC	001262	RG	#6-421	
$$PAR	000074	RG	#6-297	
$$PCHA	000746	RG	#6-332	
$$PDIN	001264	RG	#6-422	*8-550
$$PRI	000230	RG	#6-322	
$$PRM	000004	RG	#6-225	
$$SCSR	000402	RG	#6-329	
$$SNUM	000236	RG	#6-325	
$$TASK	000006	RG	#6-226	
$$TBL	000102	RG	#6-299	
$$VECT	000232	RG	#6-323	
.ALLOC =	000526	RG	#6-376	
.AMAXC	000052	RG	#6-282	
.AMAXH	000050	RG	#6-281	
.BFPTR =	000450	RG	#6-357	6-358
.CCBNM =	000452	RG	#6-358	6-359
.CCBSZ =	000454	RG	#6-359	6-360
.FUDGE =	000524	RG	#6-375	6-376
.HOME =	000446	RG	#6-356	6-357
.MAXC	000034	RG	#6-275	
.MAXH	000036	RG	#6-276	
.MXPRO =	000470	RG	#6-365	6-366
.NA	000042	RG	#6-278	
.NAMBF =	000504	RG	#6-369	6-370
.NBEA	000046	RG	#6-280	
.NBRA	000044	RG	#6-279	
.NN	000032	RG	#6-274	
.NOMAP =	000514	RG	#6-371	6-372
.NTPSZ =	000472	RG	#6-366	6-367
.PCB =	000512	RG	#6-370	6-371
.PNAM1 =	000474	RG	#6-367	6-368
.PNAM2 =	000500	RG	#6-368	6-369
.PUMR =	000522	RG	#6-374	6-375
.RDBNM =	000456	RG	#6-360	6-361
.RDBSZ =	000460	RG	#6-361	6-362
.RDBTH =	000466	RG	#6-364	6-365
.ROUTB	000054	RG	#6-283	
.ROUTM	000040	RG	#6-277	
.RPRI	000056	RG	#6-284	
.SDBNM =	000462	RG	#6-362	6-363
.SDBSZ =	000464	RG	#6-363	6-364
.UMRBA =	000520	RG	#6-373	6-374

NTLRQB
CEXLD -

288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 9

NTLRQB
PDVA -

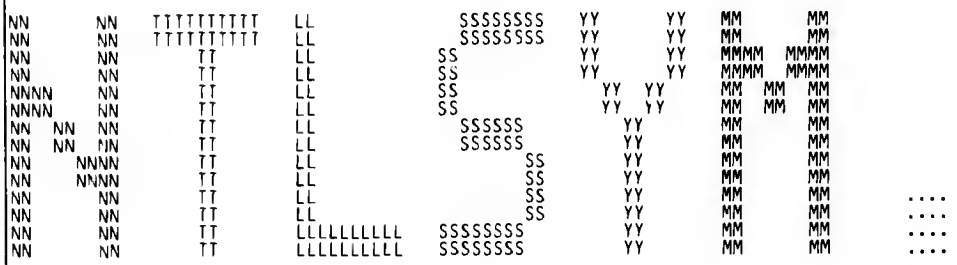
```
                                        .SBTTL  CEXLD - CHECK THAT COMM EXEC IS LOADED

                                ;+
                                ; CEXLD - MAKE SURE COMM EXEC IS LOADED
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-

000164                          CEXLD:
000164  032777  000000G 000000G         BIT     #FE.CEX,@FMASK  ; IS COMM EXEC LOADED ?
000172  001401                          BEQ     101$            ; IF EQ, NO
000174                                  RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000176                          101$:   EMSG$R  98              ; COMM EXEC NOT LOADED
```

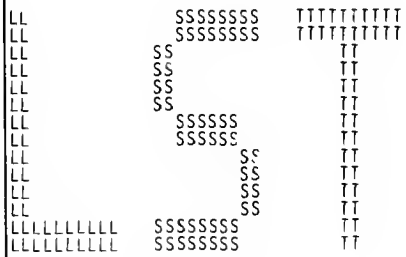
```
                                        .SBTTL  CLIN - CHECK CLEAR LINE REQUEST BLOCK

                                ;+
                                ;
                                ; CLIN - CHECK "CLEAR LINE" REQUEST BLOCK
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;
                                ;-

002054  016700  000000G         CLIN:   MOV     $SLTA,R0        ; GET SLT ADDRESS
002060  062700  000000                  ADD     #L.FLG,R0       ; POINT AT STATUS WORD
002064  032710  040000                  BIT     #LF.RDY,(R0)    ; IS LINE READY?
002070  001002                          BNE     10$             ; IF NE, YES
002072  005004                          CLR     R4              ; INDICATE NO WORK TO DO
002074  000405                          BR      20$
002076  032710  100000          10$:    BIT     #LF.ACT,(R0)    ; IS LINE ACTIVE?
002102  001003                          BNE     101$            ; IF NE, YES - NOT ALLOWED
002104  052710  020000                  BIS     #LF.UNL,(R0)    ; INDICATE LINE MARKED FOR UNLOAD
002110                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
002112                          101$:   EMSG$R  Z6              ; LINE STILL ACTIVE
```

```
                                        .SBTTL  DEV7 - REPLACE REQUESTING TASK'S NAME AND DEVICE NAME

                                ;+
                                ; DEV7 - REPLACE REQUESTING TASK'S NAME AND RSX11M DEVICE NAME
                                ;
                                ; INPUTS:
                                ;       $$TASK=REQUESTING TASK'S NAME
                                ;       R1=DEVICE NAME IN ASCII
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       R3,R4=UPDATED
                                ;-
003532  010146                  DEV7:   MOV     R1,-(SP)        ; SAVE DEVICE NAME
003534                                  CALL    REP7            ; REPLACE REQUESTING TASK'S NAME
003540  010500                          MOV     R5,R0           ; COPY ERROR MESSAGE BLOCK ADDRESS
003542  005720                          TST     (R0)+           ; GIVING ERROR TEXT ADDRESS
003544  122720  000072          10$:    CMPB    #':,(R0)+       ; LOOK FOR THE COLON
003550  001375                          BNE     10$             ;       ...
003552  005300                          DEC     R0              ; SKIP THE COLON
003554  116640  000001                  MOVB    1(SP),-(R0)     ; DEPOSIT DEVICE NAME
003560  112640                          MOVB    (SP)+,-(R0)     ;       ...
003562                                  RETURN
```

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  DIR$,QIOW$,ALUN$S,QIOW$S

                                .IF DF  R$$MPL

                                .MCALL  OLRDF$

                                OLRDF$                  ; DEFINE ONLINE RECONFIGURATIONS BITS

                                .ENDC

                        ;***
                        ; LOCAL DATA
                        ;***

000000                          .PSECT  DATA,D

                                .NLIST  BEX
                                .IF DF  R$$MPL
                        ;
                        ; ONLINE RECONFIGURATION I/O STATUS BLOCK
                        ;
                        STATUS: .BLKW   2
                                .ENDC


                        ;
                        ; TI: OUTPUT DPB
                        ;
000000                  TIOUT:  QIOW$   IO.WBT,2,2,,,,<,,40>

                        ;
                        ; POINTER INTO FILE NAME (USED FOR OPEN FAILURE ERROR MESSAGE LINE 2)
                        ;
000030                  FNPTR:  .BLKW   1

                        ;
                        ; DEFAULT UIC
                        ;
000032                  $DFUIC::.IF DF  R$$11M
                                .IF DF  M$$MGE
000032  123  131  072           .ASCIZ  "SY:[131,54]"
000046                          .BLKB   4
                                .IFF
                                .ASCIZ  "SY:[131,50]"
                                .BLKB   4
                                .ENDC
                                .IFF
                                .ASCIZ  "SY:[11,66]"
                                .BLKB   5
                                .ENDC

                        ;
                        ; CONFIG FILE UIC
                        ;
```

FILEID**NTLSYM





NTLSYM - OFFSET SYMBOL DEFINITI MACRO v05.03b Saturday 29-Jun-85 00:34 Page 4

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  EMSG$,EMSG$R,GCL$,NTLER$

                        ;***
                        ; LOCAL DATA
                        ;***

000000                          .PSECT  DATA,D

                        ;
                        ; TEMPLATE FILE FLAGS
                        ;
000000                  $FLAG1::.BLKW   1

000001                  F1.BBO==1               ; BINARY BUFFER OVERFLOWED
000002                  F1.INT==2               ; ".INT" OPERATOR SEEN
000004                  F1.IN1==4               ; ".INT1" OPERATOR SEEN
000010                  F1.IN2==10              ; ".INT2" OPERATOR SEEN
000020                  F1.IN3==20              ; ".INT3" OPERATOR SEEN
000040                  F1.LFI==40              ; ".LFILE" OPERATOR SEEN
000100                  F1.LFL==100             ; ".LFLHD" OPERATOR SEEN
000200                  F1.MPL==200             ; ".MPLHD" OPERATOR SEEN
000400                  F1.MPT==400             ; ".MPTAB" OPERATOR SEEN
001000                  F1.MXT==1000            ; ".MXTAB" OPERATOR SEEN
002000                  F1.STE==2000            ; SYMBOL TABLE ERROR HAS OCCURRED
004000                  F1.UMR==4000            ; ALLOCATE FROM UMR MAPPED AREA OF NTPOOL
010000                  F1.VSE==10000           ; VECTOR SETUP ERROR OCCURRED
020000                  F1.VC1==20000           ; VECTORS WERE SETUP
100000                  F1.ERR==100000          ; SOME ERROR OCCURRED
                                                ; *** F1.ERR MUST BE THE SIGN BIT ***

                        ;
                        ; TEMPLATE FILE LINE FLAGS
                        ;
000002                  $FLAG2::.BLKW   1

000001                  F2.BYT==1               ; BYTE MODE
000002                  F2.END==2               ; ".END" OPERATOR SEEN
000004                  F2.FIL==4               ; VALID FILE NAME SEEN
000010                  F2.LBR==10              ; LIBRARY REFERENCE ALREADY EXISTS
000020                  F2.NEG==20              ; NEGATIVE EXPRESSION
000040                  F2.SUB==40              ; NEGATIVE MODIFICATION

                        ;
                        ; BINARY BUFFER INFORMATION
                        ;
000004                  $BIN::  .BLKW   1       ; STARTING ADDRESS
000006                  $PTR::  .BLKW   1       ; CURRENT POINTER
000010                  $LEFT:: .BLKW   1       ; BYTES REMAINING

                        ;
                        ; LINE TABLE INFORMATION
                        ;
000012                  $SLTSV::.BLKW   1       ; SLT ADDRESS
000014                  $SIZE:: .BLKW   1       ; NUMBER OF BYTES NEEDED
```



```
NTLTMP
Symbol

A$$CHK
A$$CPS
A$$PRI
A$$TRP
C$$CKP
C$$ORE
C$$RSH
D$$BUG
D$$ISK
D$$L11
D$$YNC
D$$YNM
EPFX
ERRBUF
ERRMSG
ERR$10
ERR$11
ERR$12
ERR$9
E$$XPR
FL.DDN
FL.DLG
FL.ECH
FL.LLG
FMT2
FM.2
F$$LVL
F1.BBO
F1.ERR
F1.INT
F1.IN1
F1.IN2
F1.IN3
F1.LFI
F1.LFL
F1.MPL
F1.MPT
F1.MXT

. ABS.

DATA
Errors

*** A

Work
Work
Size
Size
Opera

Elaps
SY:NTL
```

NTLTMP - TEMPLATE PROCESSING MACRO V05.03b Saturday 29-Jun-85 00:34 Page 15-1
Symbol table

```
A$$CHK= 000000          F1.STE= 002000 G        N$$MLL= 000001          $ELINE  000332RG        $RQ4    000362RG
A$$CPS= 000000          F1.UMR= 004000 G        N$$MOV= 000010          $ERR13  000514R     002 $RQ5    000354RG
A$$PRI= 000000          F1.VCT= 020000 G        N$$NCT= 000001          $ERR21  000554R     002 $RQ7    000346RG
A$$TRP= 000000          F1.VSE= 010000 G        N$$PEM= 000001          $ERR79  000602R     002 $SAVAL= ****** GX
C$$CKP= 000000          F2.BYT= 000001 G        P$$P45= 000000          $ERR82  000630R     002 $SIZE   000014RG    002
C$$ORE= 000400          F2.END= 000002 G        P$$WRD= 000000          $ERR83  000676R     002 $SLTA = ****** GX
C$$RSH= 177564          F2.FIL= 000004 G        Q$$OPT= 000010          $EVEN   000470RG        $SLTSV  000012RG    002
D$$BUG= 177514          F2.LBR= 000010 G        R$$DER= 000000          $FILEN  000100RG    002 $SZPTR  000032RG    002
D$$ISK= 000000          F2.NEG= 000020 G        R$$K11= 000001          $FILEX= ****** GX       $TERR   000646RG
D$$L11= 000001          F2.SUB= 000040 G        R$$SND= 000000          $FLAGS= ****** GX       $TIOUT= ****** GX
D$$YNC= 000000          G$$TPP= 000000          R$$11M= 000000          $FLAG1  000000RG    002 $TMALT= ****** GX
D$$YNM= 000000          G$$TSS= 000000          S$$WRG= 000000          $FLAG2  000002RG    002 $TMCHK= ****** GX
EPFX    000507R     002 G$$TTK= 000000          S$$YSZ= 007600          $FNOUT= ****** GX       $TMEXP= ****** GX
ERRBUF  000361R     002 G$$WRD= 000000          TMPBF   000146R     002 $FNR50  000130RG    002 $TMFIL  000056R
ERRMSG  000354R     002 IE.EOF= ****** GX       TMPSZ   000352R     002 $GCLEP= ****** GX       $TMFIX  000306RG
ERR$10= ****** GX       IE.RBG= ****** GX       TM.RD   000210R         $ICBAD  000072RG    002 $TMLTA= ****** GX
ERR$11= ****** GX       I$$RAR= 000000          T$$KMG= 000000          $ICBSZ  000074RG    002 $TMP    000000RG
ERR$12= ****** GX       I$$RDN= 000000          T$$MIN= 000000          $JSROF= 000012 G        $TMPKW= ****** GX
ERR$9 = ****** GX       K$$CNT= 177546          V$$CTR= 001000          $LBADR  000140RG    002 $TMPST= ****** GX
E$$XPR= 000000          K$$CSR= 177546          WPFX    000472R     002 $LBIAS  000144RG    002 $TMVCT= ****** GX
FL.DDM= ****** GX       K$$LDC= 000000          X$$DBT= 000000          $LDA    000054RG    002 $TM.IN= ****** GX
FL.DLC= ****** GX       K$$TPS= 000074          $ABFI = ****** GX       $LEFT   000010RG    002 $TPARS= ****** GX
FL.ECH= ****** GX       LD$LP = 000000          $ADDR   000016RG    002 $LFLD   000060RG    002 $TWARN  000622RG
FL.LLC= ****** GX       L$$ASG= 000000          $BFALL  000114RG    002 $MPLD   000056RG    002 $TWE    000636RG
FMT2    000740R     002 L$$DRV= 000000          $BFBAS  000126RG    002 $MXADD  000020RG    002 $UBIAS  000050RG    002
FM.2  = 000000          L$$P11= 000001          $BFLBN  000116RG    002 $MXLEN  000026RG    002 $UDA    000052RG    002
F$$LVL= 000001          L$$11R= 000000          $BFPTR  000122RG    002 $MXSIZ  000022RG    002 $VALUE  000076RG    002
F1.BBO= 000001 G        M$$CRB= 000124          $BFXFR  000124RG    002 $NALL   000030RG    002 $VECT1  000062RG    002
F1.ERR= 100000 G        M$$CRX= 000000          $BIN    000004RG    002 $NLIB   000031RG    002 $VECT2  000064RG    002
F1.INT= 000002 G        M$$FCS= 000000          $CAPR   000142RG    002 $NVECT  000070RG    002 $VECT3  000066RG    002
F1.IN1= 000004 G        M$$MGE= 000000          $CLBUF= ****** GX       $OFF    000024RG    002 $VFDDM  000522RG
F1.IN2= 000010 G        M$$NET= 000000          $CLDEQ= ****** GX       $PRAPR= 120000 G        $VFDLC  000542RG
F1.IN3= 000020 G        M$$OVR= 000000          $CLIOS= ****** GX       $PTR    000006RG    002 $VFLV1  000574RG
F1.LFI= 000040 G        N$$ACC= 000001          $CLSIZ= ****** GX       $REP.C= ****** GX       $VFSPC  000562RG
F1.LFL= 000100 G        N$$BUF= 000001          $CVAL   000136RG    002 $RQ1    000404RG        .ABFI   000316RG
F1.MPL= 000200 G        N$$LDV= 000001          $DIFF   000112RG    002 $RQ2    000376RG        .BYTE = ****** GX
F1.MPT= 000400 G        N$$MCP= 000001          $EFNLL  000326RG        $RQ3    000370RG        .TPARS= ****** GX
F1.MXT= 001000 G

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000754    001   (RW,I,LCL,REL,CON)
DATA    000760    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10418 Words  ( 41 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:15.41
SY:NTLTMP.V2,[132,134]NTLTMP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[ 30,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLTMP
```


NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES					
$NLDEV	000000	RG	#8-239					
$PCB	= ******	GX	8-259					
$PRV.N	= ******	GX	7-157	7-158	7-161			
$READ	= ******	GX	13-539					
$REP.C	= ******	GX	7-162	7-164				
$REP.P	= ******	GX	7-159	7-160	18-765			
$RLBL	= ******	GX	11-446					
$SAVAL	= ******	GX	16-699					
$SYMBL	= ******	GX	10-389					
$TIOUT	= ******	GX	16-716					
$$DEV	= ******	GX	9-324	10-371	10-372	13-542	19-810	19-811
$$TBL	= ******	GX	8-257					

MACRO CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES					
.TTO	000000	RG	#8-240					
ASTT	= ******	GX	8-258					
CVDAT	= ******	GX	8-260					
SX$OK	= ******	GX	8-261					
TSPC	= ******	GX	8-262					
$$MPL	= ******	GX	8-263					
$$11D	= ******	GX	8-264					
$$11M	= ******	GX	8-265					
$$11S	= ******	GX	8-266					
.VXTN	= ******	GX	8-267					
RSTD	= ******	GX	8-268					
$$BAS	= ******	GX	8-269					
.DATN	= ******	GX	8-270					
.SCB	= ******	GX	8-271					
CFUIC	= ******	GX	8-272					
CRAVL	= ******	GX	8-273					
CXUIC	= ******	GX	8-274					
DFUIC	= ******	GX	8-275					
ERR40	= ******	GX	8-276					
FLAGS	= ******	GX	8-277					
FUNC	= ******	GX	8-278					
HEADR	= ******	GX	8-279					
LDUIC	= ******	GX	8-280					
NLGET	= ******	GX	8-281					
NLINI	= ******	GX	8-282					
NLPUT	= ******	GX	8-283					
RERR	= ******	GX	8-284					
TTO	= ******	GX	8-285					
XBUF2	= ******	GX	8-286					
XLEN2	= ******	GX	8-287					
XXBUF	= ******	GX	8-288					
XXINI	= ******	GX	8-289					
XXLEN	= ******	GX	8-290					
$AST	= ******	GX	8-291					
$LRB	= ******	GX	8-292					
$PRM	= ******	GX	8-293					
$TASK	= ******	GX	8-294					
$$GLB	= ******	GX	8-295					
$$T1	= ******	GX	8-296					

SYMBOL C

SYMBOL

NTLGET CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
QASTT	= ****** GX	9-249							
RCVDAT	000000 R	#5-81	7-168						
RSX$OK	= 000000	#6-126	6-126						
RTSPC	000234 R	5-99	#7-180						
R$$MPL	= ******	6-125							
R$$11D	= ******	5-62							
R$$11M	= 000000	5-98							
R$$11S	= ******	6-126							
R.VXTN	= 000002	5-81							
SRSTD	= ****** GX	9-230							
S$$BAS	= ******	5-99	5-99						
S.DATN	= 000002	5-86	*8-200	*8-201					
U.SCB	= ****** GX	6-126							
$CFUIC	= ****** GX	6-133							
$CRAVL	= ****** GX	6-126							
$CXUIC	= ****** GX	6-131							
$DFUIC	= ****** GX	6-132							
$ERR40	000064 R	#5-99	6-150						
$FLAGS	= ****** GX	*6-143	6-144	*6-144	7-171	*7-171	*7-179		
$FUNC	= ****** GX	*6-151	8-203	9-246					
$HEADR	= ****** GX	9-236							
$LDUIC	= ****** GX	6-130							
$NLGET	000146 RG	6-152	#7-167						
$NLINI	000000 RG	#6-124							
$NLPUT	000236 RG	6-153	#8-193						
$RERR	= ****** GX	5-99							
$TTO	= ****** GX	6-126							
$XBUF2	= ****** GX	*6-141							
$XLEN2	= ****** GX	*6-142							
$XXBUF	= ****** GX	6-141							
$XXINI	= ****** GX	6-140							
$XXLEN	= ****** GX	6-142							
$$AST	= ****** GX	*7-173	*7-176	8-195	9-243				
$$LRB	000022 RG	5-81	5-86	#5-91	7-169	7-170	7-174	7-176	7-177	8-202
$$PRM	= ****** GX	*7-177	9-245						
$$TASK	= ****** GX	*7-169	*7-170	8-200	8-201	9-229			
$$$GLB	= ******	5-81	5-86						
$$$T1	= 000006	#5-81	5-81	#5-86	5-86	5-86	5-86		
.TTO	= ****** GX	6-126							

NTLKMC
KMC LOA

366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386

NTLGET CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

NTLKMC
KMC LOA

```
                                        CMP     (R1)+,(R1)+             ;U-SKIP TO NEXT ENTRY
                                        CALL    KMCLD                   ;U-LOAD UP THE KMC
                                        BR      30$                     ;U-AND COUNT THE ENTRY
                                20$:    ADD     #6,R1                   ;U-SKIP EMPTY ENTRIES (4 WORDS/ENTRY)
                                30$:    DEC     R2                      ;U-COUNT DOWN
                                        BGT     10$                     ;U-BR IF MORE ENTRIES
                                        BR      5$                      ;U-GO CHECK IF MORE BLOCKS IN LIST

                                        .ENDC                           ;*-MAPPED AND UNMAPPED TAKE SAME EXIT-*

000230  000241                    40$:    CLC                             ;B-NO ERRORS FOR NOW
000232                                  RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
                                        .IF DF  R$$MPL
                                101$:   EMSG$   3K                      ;M+ CAN'T FIND UNIBUS RUN MASK
                                        SEC                             ;M+ INDICATE FAILURE
                                        RETURN                          ;M+

                                        .ENDC
```

```
    909         000001                          .END
```

```
                                        .SBTTL  $XFER - TRANSFER PROCESS TO CORE
                                ;+
                                ; $XFER - TRANSFER PROCESS FROM DISK TO CORE
                                ;
                                ; INPUTS:
                                ;       $PDVA=PDV ADDRESS
                                ;       $NAME=RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0 = PDV ADDRESS
                                ;       R1 = PCB ADDRESS
                                ;       $BIAS=PROCESS RELOCATION BIAS
                                ;-
000000                          $XFER::
000000  016700  000000G                 MOV     $PDVA,R0        ; GET PDV ADDRESS
000004  005760  000000G                 TST     Z.PCB(R0)       ; IS PROCESS ALREADY CORE-RESIDENT?
000010  001102                          BNE     10$             ; IF NE, YES - NETLDR LOGIC ERROR

000012  016700  000000G                 MOV     $NAME,R0        ; GET PDV NAME
000016                                  CALL    $DFN            ; SETUP DEFAULT FILE NAME
000022                                  CALL    PRSYM           ; LOOK FOR SPECIAL SYMBOLS
000026  103472                          BCS     20$             ; IF CS, FAILURE
000030  005000                          CLR     R0              ; SETUP ASCIZ FILE SPEC
000032  005001                          CLR     R1              ; FOR [131,5X]XXX.TSK

000034  005002                          CLR     R2              ;       ...
000036                                  CALL    $AZFS           ;       ...
000042                                  CALL    $PRID           ; TRANSFER PROCESS INTO CORE
000046  103467                          BCS     TSKERR          ; IF CS, ERROR
000050  016700  000000G                 MOV     $PDVA,R0        ; GET PDV ADDRESS

000054  010167  000000G                 MOV     R1,$PCB         ; SAVE PCB ADDRESS FOR LATER
000060  016160  000000G 000000G         MOV     P.REL(R1),Z.DSP(R0) ; SET PROCESS RELOCATION BIAS
000066  016167  000000G 000000G         MOV     P.REL(R1),$BIAS ; TWICE
000074  016760  000030' 000002G         MOV     PVAL,Z.DSP+2(R0) ; SET DISPATCH TABLE ADDRESS
000102  016703  000000G                 MOV     $LDSIZ,R3       ; GET LOAD SIZE OF PROCESS
000106  026103  000000G                 CMP     P.BLKS(R1),R3   ; LOAD SIZE SAME AS PARTITION SIZE?
000112  001001                          BNE     5$              ; IF NE, NO .. THERE MUST BE SOME SPACE
000114  005003                          CLR     R3              ; ELSE, SAY THERE IS NO LISTHEAD

000116                          5$:     ASL$    6,R3            ; GET LOAD SIZE IN BYTES
000132  010360  000000G                 MOV     R3,Z.AVL(R0)    ; INITIALIZE LISTHEAD
000136  001423                          BEQ     10$             ; IF EQ, NO LIST HEAD
000140  016104  000000G                 MOV     P.BLKS(R1),R4   ; GET PARTITION SIZE IN BLOCKS
000144                                  ASL$    6,R4            ; CONVERT TO BYTES
000160  160304                          SUB     R3,R4           ; CALCULATE SIZE OF FREE SPACE

000162                                  SWSTK$  10$             ;* ENTER KERNEL MODE
000166  016700  000000G                 MOV     $BIAS,R0        ;* BIAS OF PARTITION
000172                                  CALL    $MAPX           ;* MAP TO PARTITION
000176  050302                          BIS     R3,R2           ;* GET ADDRESS OF START OF FREE SPACE IN R2
000200  005022                          CLR     (R2)+           ;* INITIALIZE LINK POINTER
000202  010412                          MOV     R4,(R2)         ;* INITIALIZE FREE SPACE BYTE COUNT
000204                                  RETURN                  ;* BACK TO USER MODE

000206                          10$:    CALL    AXSYM           ; FILL IN SPECIAL SYMBOLS IF PROCESS IS AUX
```



```
NTLPRO
Symbol

T$NLEN
T$NNUL
T$NOPL
T$NRNI
T$NRPL
T$NRUL
T$NVR
T$RPRI
T$SVC
T$T5
T$T6
T$$KMG
T$$MIN
T1SCL
T100C
V$$CTR
XBLK1

. ABS.

DATA
Errors

*** As

Work
Work
Size o
Size o
Operat

Elapse
SY:NTL
```

NTLPRO - "SET PROCESS" ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:31 Page 13-3
Symbol table

```
T$NLEN  000010          XBLK2   000140R     002 $DFN  = ****** GX      $ERR60= ****** GX       $PCB  = ****** GX
T$NNUL  000002          XBLK3   000152R     002 $DFNAM= ****** GX      $ERR61= ****** GX       $PDVA = ****** GX
T$NOPL  000006          XPTPDV  000002R     002 $ERRQX= ****** GX      $ERR62= ****** GX       $PRCLN  000054RG    002
T$NRNI  000042          X$$DBT= 000000          $ERR44= ****** GX      $ERR63= ****** GX       $PREXT  000044RG    002
T$NRPL  000005          Z.AVL = ****** GX       $ERR45= ****** GX      $ERR64= ****** GX       $PRINC  000032RG    002
T$NRUL  000007          Z.DAT = ****** GX       $ERR46= ****** GX      $ERR65= ****** GX       $PRIO = ****** GX
T$NVR   000001          Z.DSP = ****** GX       $ERR47= ****** GX      $ERR66= ****** GX       $PRLLN  000042RG    002
T$RPRI  000040          Z.PCB = ****** GX       $ERR49= ****** GX      $ERR67= ****** GX       $ROL  = ****** GX
T$SVC   000034          $ADJ  = ****** GX       $ERR50= ****** GX      $ERR69= ****** GX       $SAVAL= ****** GX
T$T5    000030          $ALECL  001220RG        $ERR51= ****** GX      $ERR70= ****** GX       $SQSRV  000100R     002
T$T6    000032          $ALMHC  001022RG        $ERR53= ****** GX      $INICO  000764RG        $SYMBL= ****** GX
T$$KMG= 000000          $ALN18= ****** GX       $ERR54= ****** GX      $LDRSZ= ****** GX       $XFER   000000RG
T$$MIN= 000000          $AZFS = ****** GX       $ERR55= ****** GX      $MAPX = ****** GX       $$DEV = ****** GX
T1SCL = ****** GX       $BIAS = ****** GX       $ERR56= ****** GX      $MUL  = ****** GX       $$NODC= ****** GX
T100C = ****** GX       $CAT5 = ****** GX       $ERR57= ****** GX      $NAME = ****** GX       $$TBL = ****** GX
V$$CTR= 001000          $CKT  = ****** GX       $ERR58= ****** GX      $NICAH= ****** GX       .CXSYM= ****** GX
XBLK1   000126R     002 $CTSYM  000456RG        $ERR59= ****** GX

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001462    001   (RW,I,LCL,REL,CON)
DATA    000236    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 16470 Words  ( 65 Pages)
Size of core pool: 17600 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:23.16
SY:NTLPRO.V2,[132,134]NTLPRO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLPRO
```

NTLROO
VECTORE

```
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
20
202
203
204
205
206
207
208
209
210
211
212
213
214
215
21
```


NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 1

NTLROO
VECTORE

```
                                ; VECTORED COMM EXEC SYMBOLS (CONT'D)

000142  000000G        PDVNM:: $PDVNM
000144  000000G        SLTMA:: $SLTMA
000146  000000G        SLTNM:: $SLTNM
000150  000000G        DECPT:: $DECPT
000152  000000G        FILHD:: $FILHD
000154  000000G        LGSTT:: $LGSTT
000156  000000G        NS0::   $NS0
000160  000000G        NS1::   $NS1
000162  000000G        NS2::   $NS2
000164  000000G        NS3::   $NS3
000166  000000G        NS4::   $NS4
000170  000000G        NS5::   $NS5
000172  000000G        NS6::   $NS6
000174  000000G        NS7::   $NS7
000176  000000G        PDVID:: $PDVID
000200  000000G        LLCAL:: $LLCAL
000202  000000G        NETPF:: $NETPF
000204  000000G        LGPDV:: $LGPDV
000206  000000G        LGDDB:: $LGDDB
000210  000000G        T100Q:: $T100Q
000212  000000G        T100C:: $T100C
000214  000000G        T1SCL:: $T1SCL
000216  000000G        RDBSZ:: $RDBSZ
000220  000000G        CCBLH:: $CCBLH
000222  000000G        CCBSZ:: $CCBSZ
000224  000000G        RDBLH:: $RDBLH
000226  000000G        SDBLH:: $SDBLH
000230  000000G        CXOPT:: $CXOPT
000232  000000G        NTLPT:: $NTLPT
000234  000000G        OBJHD:: $OBJHD
000236  000000G        PUMR::  $PUMR

                                .IF NDF R$$MPL
000240  000000G        CEAVL:: $CEAVL
                                .ENDC

000242  000000G        STMFC:: $STMFC
000244  000000G        PAVL::  $XAVL
000246  000000G        XBIAS:: $XBIAS
000250  000000G        XAVL::  $XAVL
000252  000000G        QBIAS:: $QBIAS
000254  000000G        QSTRT:: $QSTRT
000256  000000G        RDBNM:: $RDBNM
000260  000000G        SDBNM:: $SDBNM
000262  000000G        SDBSZ:: $SDBSZ
000264  000000G        ALOC1:: $ALOC1
000266  000000G        ALOCB:: $ALOCB
000270  000000G        HEADR:: $HEADR
000272  000000G        PSIPT:: $PSIPT

        000136          $VECLN == <<.-$VECBG>/2>

000000                          .PSECT  DATA,D
```



```
                        .ENABL  LSB
000000  000000G  $VECBG::  .WORD  $VECBG
000002  000000G  $VECEX::  .WORD  $VECEX
000004  000000G  $SCIR::   .WORD  $SCIR
000006  000000G  $SCEX::   .WORD  $SCEX
000010  000000G  $SCSR::   .WORD  $SCSR
000012  000000G  $SCSX::   .WORD  $SCSX
000014  000000G  $SCSV::   .WORD  $SCSV
000016  000000G  $SCSD::   .WORD  $SCSD
000020  000000G  $SCSP::   .WORD  $SCSP
000022  000000G  $SCSC::   .WORD  $SCSC
000024  000000G  $SCSA::   .WORD  $SCSA
000026  000000G  $SCSB::   .WORD  $SCSB
000030  000000G  $SCSE::   .WORD  $SCSE
000032  000000G  $SCSF::   .WORD  $SCSF
000034  000000G  $SCSG::   .WORD  $SCSG
000036  000000G  $SCSH::   .WORD  $SCSH
000040  000000G  $SCSI::   .WORD  $SCSI
000042  000000G  $SCSJ::   .WORD  $SCSJ
000044  000000G  $SCSK::   .WORD  $SCSK
000046  000000G  $SCSL::   .WORD  $SCSL
000050  000000G  $SCSM::   .WORD  $SCSM
000052  000000G  $SCSN::   .WORD  $SCSN
000054  000000G  $SCSO::   .WORD  $SCSO
000056  000000G  $SCSQ::   .WORD  $SCSQ
000060  000000G  $SCSS::   .WORD  $SCSS
000062  000000G  $SCST::   .WORD  $SCST
000064  000000G  $SCSU::   .WORD  $SCSU
000066  000000G  $SCSW::   .WORD  $SCSW
000070  000000G  $SCSX::   .WORD  $SCSX
000072  000000G  $SCSY::   .WORD  $SCSY
000074  000000G  $SCSZ::   .WORD  $SCSZ
000076  000000G  $SCS0::   .WORD  $SCS0
000100  000000G  $SCS1::   .WORD  $SCS1
000102  000000G  $SCS2::   .WORD  $SCS2
000104  000000G  $SCS3::   .WORD  $SCS3
000106  000000G  $SCS4::   .WORD  $SCS4
000110  000000G  $SCS5::   .WORD  $SCS5
000112  000000G  $SCS6::   .WORD  $SCS6
000114  000000G  $SCS7::   .WORD  $SCS7
000116  000000G  $SCS8::   .WORD  $SCS8
000120  000000G  $SCS9::   .WORD  $SCS9
000122  000000G  $SCSR0::  .WORD  $SCSR0
000124  000000G  $SCSR1::  .WORD  $SCSR1
000126  000000G  $SCSR2::  .WORD  $SCSR2
000130  000000G  $SCSR3::  .WORD  $SCSR3
000132  000000G  $SCSR4::  .WORD  $SCSR4
000134  000000G  $SCSR5::  .WORD  $SCSR5
000136  000000G  $SCSR6::  .WORD  $SCSR6
000140  000000G  $SCSR7::  .WORD  $SCSR7
000142  000000G  $SCSR8::  .WORD  $SCSR8
000144  000000G  $SCSR9::  .WORD  $SCSR9
000146  000000G  $SCSRA::  .WORD  $SCSRA
000150  000000G  $SCSRB::  .WORD  $SCSRB
000152  000000G  $SCSRC::  .WORD  $SCSRC
000154  000000G  $SCSRD::  .WORD  $SCSRD
000156  000000G  $SCSRE::  .WORD  $SCSRE
000160  000000G  $SCSRF::  .WORD  $SCSRF
000162  000000G  $SCSRG::  .WORD  $SCSRG
000164  000000G  $SCSRH::  .WORD  $SCSRH
000166  000000G  $SCSRI::  .WORD  $SCSRI
000170  000000G  $SCSRJ::  .WORD  $SCSRJ
000172  000000G  $SCSRK::  .WORD  $SCSRK
000174  000000G  $SCSRL::  .WORD  $SCSRL
000176  000000G  $SCSRM::  .WORD  $SCSRM
000200  000000G  $SCSRN::  .WORD  $SCSRN
000202  000000G  $SCSRO::  .WORD  $SCSRO
000204  000000G  $SCSRP::  .WORD  $SCSRP
000206  000000G  $SCSRQ::  .WORD  $SCSRQ
000210  000000G  $SCSRR::  .WORD  $SCSRR
000212  000000G  $SCSRS::  .WORD  $SCSRS
000214  000000G  $SCSRT::  .WORD  $SCSRT
000216  000000G  $SCSRU::  .WORD  $SCSRU
000220  000000G  $SCSRV::  .WORD  $SCSRV
000222  000000G  $SCSRW::  .WORD  $SCSRW
000224  000000G  $SCSRX::  .WORD  $SCSRX
000226  000000G  $SCSRY::  .WORD  $SCSRY
000230  000000G  $SCSRZ::  .WORD  $SCSRZ
000232  000000G  $SCSR00::  .WORD  $SCSR00
000234  000000G  $SCSR01::  .WORD  $SCSR01
000236  000000G  $SCSR02::  .WORD  $SCSR02
000240  000000G  $SCSR03::  .WORD  $SCSR03
000242  000000G  $SCSR04::  .WORD  $SCSR04
000244  000000G  $SCSR05::  .WORD  $SCSR05
000246  000000G  $SCSR06::  .WORD  $SCSR06
000250  000000G  $SCSR07::  .WORD  $SCSR07
000252  000000G  $SCSR08::  .WORD  $SCSR08
000254  000000G  $SCSR09::  .WORD  $SCSR09
000256  000000G  $SCSR0A::  .WORD  $SCSR0A
000260  000000G  $SCSR0B::  .WORD  $SCSR0B
000262  000000G  $SCSR0C::  .WORD  $SCSR0C
000264  000000G  $SCSR0D::  .WORD  $SCSR0D
000266  000000G  $SCSR0E::  .WORD  $SCSR0E
000270  000000G  $SCSR0F::  .WORD  $SCSR0F
000272  000000G  $SCSR0G::  .WORD  $SCSR0G
000274  000000G  $SCSR0H::  .WORD  $SCSR0H
000276  000000G  $SCSR0I::  .WORD  $SCSR0I
000300  000000G  $SCSR0J::  .WORD  $SCSR0J
000302  000000G  $SCSR0K::  .WORD  $SCSR0K
000304  000000G  $SCSR0L::  .WORD  $SCSR0L
000306  000000G  $SCSR0M::  .WORD  $SCSR0M
000310  000000G  $SCSR0N::  .WORD  $SCSR0N
000312  000000G  $SCSR0O::  .WORD  $SCSR0O
000314  000000G  $SCSR0P::  .WORD  $SCSR0P
000316  000000G  $SCSR0Q::  .WORD  $SCSR0Q
000320  000000G  $SCSR0R::  .WORD  $SCSR0R
000322  000000G  $SCSR0S::  .WORD  $SCSR0S
000324  000000G  $SCSR0T::  .WORD  $SCSR0T
000326  000000G  $SCSR0U::  .WORD  $SCSR0U
000330  000000G  $SCSR0V::  .WORD  $SCSR0V
000332  000000G  $SCSR0W::  .WORD  $SCSR0W
000334  000000G  $SCSR0X::  .WORD  $SCSR0X
000336  000000G  $SCSR0Y::  .WORD  $SCSR0Y
000340  000000G  $SCSR0Z::  .WORD  $SCSR0Z
000342  000000G  $SCSR10::  .WORD  $SCSR10
000344  000000G  $SCSR11::  .WORD  $SCSR11
000346  000000G  $SCSR12::  .WORD  $SCSR12
000350  000000G  $SCSR13::  .WORD  $SCSR13
000352  000000G  $SCSR14::  .WORD  $SCSR14
000354  000000G  $SCSR15::  .WORD  $SCSR15
000356  000000G  $SCSR16::  .WORD  $SCSR16
000360  000000G  $SCSR17::  .WORD  $SCSR17
000362  000000G  $SCSR18::  .WORD  $SCSR18
000364  000000G  $SCSR19::  .WORD  $SCSR19
000366  000000G  $SCSR1A::  .WORD  $SCSR1A
000370  000000G  $SCSR1B::  .WORD  $SCSR1B
000372  000000G  $SCSR1C::  .WORD  $SCSR1C
000374  000000G  $SCSR1D::  .WORD  $SCSR1D
000376  000000G  $SCSR1E::  .WORD  $SCSR1E
000400  000000G  $SCSR1F::  .WORD  $SCSR1F
000402  000000G  $SCSR1G::  .WORD  $SCSR1G
000404  000000G  $SCSR1H::  .WORD  $SCSR1H
000406  000000G  $SCSR1I::  .WORD  $SCSR1I
000410  000000G  $SCSR1J::  .WORD  $SCSR1J
000412  000000G  $SCSR1K::  .WORD  $SCSR1K
000414  000000G  $SCSR1L::  .WORD  $SCSR1L
000416  000000G  $SCSR1M::  .WORD  $SCSR1M
000420  000000G  $SCSR1N::  .WORD  $SCSR1N
000422  000000G  $SCSR1O::  .WORD  $SCSR1O
000424  000000G  $SCSR1P::  .WORD  $SCSR1P
000426  000000G  $SCSR1Q::  .WORD  $SCSR1Q
000430  000000G  $SCSR1R::  .WORD  $SCSR1R
000432  000000G  $SCSR1S::  .WORD  $SCSR1S
000434  000000G  $SCSR1T::  .WORD  $SCSR1T
000436  000000G  $SCSR1U::  .WORD  $SCSR1U
000440  000000G  $SCSR1V::  .WORD  $SCSR1V
000442  000000G  $SCSR1W::  .WORD  $SCSR1W
000444  000000G  $SCSR1X::  .WORD  $SCSR1X
000446  000000G  $SCSR1Y::  .WORD  $SCSR1Y
000450  000000G  $SCSR1Z::  .WORD  $SCSR1Z
000452  000000G  $SCSR20::  .WORD  $SCSR20
000454  000000G  $SCSR21::  .WORD  $SCSR21
000456  000000G  $SCSR22::  .WORD  $SCSR22
000460  000000G  $SCSR23::  .WORD  $SCSR23
000462  000000G  $SCSR24::  .WORD  $SCSR24
000464  000000G  $SCSR25::  .WORD  $SCSR25
000466  000000G  $SCSR26::  .WORD  $SCSR26
000470  000000G  $SCSR27::  .WORD  $SCSR27
000472  000000G  $SCSR28::  .WORD  $SCSR28
000474  000000G  $SCSR29::  .WORD  $SCSR29
000476  000000G  $SCSR2A::  .WORD  $SCSR2A
000500  000000G  $SCSR2B::  .WORD  $SCSR2B
000502  000000G  $SCSR2C::  .WORD  $SCSR2C
000504  000000G  $SCSR2D::  .WORD  $SCSR2D
000506  000000G  $SCSR2E::  .WORD  $SCSR2E
000510  000000G  $SCSR2F::  .WORD  $SCSR2F
000512  000000G  $SCSR2G::  .WORD  $SCSR2G
000514  000000G  $SCSR2H::  .WORD  $SCSR2H
000516  000000G  $SCSR2I::  .WORD  $SCSR2I
000520  000000G  $SCSR2J::  .WORD  $SCSR2J
000522  000000G  $SCSR2K::  .WORD  $SCSR2K
000524  000000G  $SCSR2L::  .WORD  $SCSR2L
000526  000000G  $SCSR2M::  .WORD  $SCSR2M
000530  000000G  $SCSR2N::  .WORD  $SCSR2N
000532  000000G  $SCSR2O::  .WORD  $SCSR2O
000534  000000G  $SCSR2P::  .WORD  $SCSR2P
000536  000000G  $SCSR2Q::  .WORD  $SCSR2Q
000540  000000G  $SCSR2R::  .WORD  $SCSR2R
000542  000000G  $SCSR2S::  .WORD  $SCSR2S
000544  000000G  $SCSR2T::  .WORD  $SCSR2T
000546  000000G  $SCSR2U::  .WORD  $SCSR2U
000550  000000G  $SCSR2V::  .WORD  $SCSR2V
000552  000000G  $SCSR2W::  .WORD  $SCSR2W
000554  000000G  $SCSR2X::  .WORD  $SCSR2X
000556  000000G  $SCSR2Y::  .WORD  $SCSR2Y
000560  000000G  $SCSR2Z::  .WORD  $SCSR2Z
000562  000000G  $SCSR30::  .WORD  $SCSR30
000564  000000G  $SCSR31::  .WORD  $SCSR31
000566  000000G  $SCSR32::  .WORD  $SCSR32
000570  000000G  $SCSR33::  .WORD  $SCSR33
000572  000000G  $SCSR34::  .WORD  $SCSR34
000574  000000G  $SCSR35::  .WORD  $SCSR35
000576  000000G  $SCSR36::  .WORD  $SCSR36
000600  000000G  $SCSR37::  .WORD  $SCSR37
000602  000000G  $SCSR38::  .WORD  $SCSR38
000604  000000G  $SCSR39::  .WORD  $SCSR39
000606  000000G  $SCSR3A::  .WORD  $SCSR3A
000610  000000G  $SCSR3B::  .WORD  $SCSR3B
000612  000000G  $SCSR3C::  .WORD  $SCSR3C
000614  000000G  $SCSR3D::  .WORD  $SCSR3D
000616  000000G  $SCSR3E::  .WORD  $SCSR3E
000620  000000G  $SCSR3F::  .WORD  $SCSR3F
000622  000000G  $SCSR3G::  .WORD  $SCSR3G
000624  000000G  $SCSR3H::  .WORD  $SCSR3H
000626  000000G  $SCSR3I::  .WORD  $SCSR3I
000630  000000G  $SCSR3J::  .WORD  $SCSR3J
000632  000000G  $SCSR3K::  .WORD  $SCSR3K
000634  000000G  $SCSR3L::  .WORD  $SCSR3L
000636  000000G  $SCSR3M::  .WORD  $SCSR3M
000640  000000G  $SCSR3N::  .WORD  $SCSR3N
000642  000000G  $SCSR3O::  .WORD  $SCSR3O
000644  000000G  $SCSR3P::  .WORD  $SCSR3P
000646  000000G  $SCSR3Q::  .WORD  $SCSR3Q
000650  000000G  $SCSR3R::  .WORD  $SCSR3R
000652  000000G  $SCSR3S::  .WORD  $SCSR3S
000654  000000G  $SCSR3T::  .WORD  $SCSR3T
000656  000000G  $SCSR3U::  .WORD  $SCSR3U
000660  000000G  $SCSR3V::  .WORD  $SCSR3V
000662  000000G  $SCSR3W::  .WORD  $SCSR3W
000664  000000G  $SCSR3X::  .WORD  $SCSR3X
000666  000000G  $SCSR3Y::  .WORD  $SCSR3Y
000670  000000G  $SCSR3Z::  .WORD  $SCSR3Z
000672  000000G  $SCSR40::  .WORD  $SCSR40
000674  000000G  $SCSR41::  .WORD  $SCSR41
000676  000000G  $SCSR42::  .WORD  $SCSR42
000700  000000G  $SCSR43::  .WORD  $SCSR43
000702  000000G  $SCSR44::  .WORD  $SCSR44
000704  000000G  $SCSR45::  .WORD  $SCSR45
000706  000000G  $SCSR46::  .WORD  $SCSR46
000710  000000G  $SCSR47::  .WORD  $SCSR47
000712  000000G  $SCSR48::  .WORD  $SCSR48
000714  000000G  $SCSR49::  .WORD  $SCSR49
000716  000000G  $SCSR4A::  .WORD  $SCSR4A
000720  000000G  $SCSR4B::  .WORD  $SCSR4B
000722  000000G  $SCSR4C::  .WORD  $SCSR4C
000724  000000G  $SCSR4D::  .WORD  $SCSR4D
000726  000000G  $SCSR4E::  .WORD  $SCSR4E
000730  000000G  $SCSR4F::  .WORD  $SCSR4F
000732  000000G  $SCSR4G::  .WORD  $SCSR4G
000734  000000G  $SCSR4H::  .WORD  $SCSR4H
000736  000000G  $SCSR4I::  .WORD  $SCSR4I
000740  000000G  $SCSR4J::  .WORD  $SCSR4J
000742  000000G  $SCSR4K::  .WORD  $SCSR4K
000744  000000G  $SCSR4L::  .WORD  $SCSR4L
000746  000000G  $SCSR4M::  .WORD  $SCSR4M
000750  000000G  $SCSR4N::  .WORD  $SCSR4N
000752  000000G  $SCSR4O::  .WORD  $SCSR4O
000754  000000G  $SCSR4P::  .WORD  $SCSR4P
000756  000000G  $SCSR4Q::  .WORD  $SCSR4Q
000760  000000G  $SCSR4R::  .WORD  $SCSR4R
000762  000000G  $SCSR4S::  .WORD  $SCSR4S
000764  000000G  $SCSR4T::  .WORD  $SCSR4T
000766  000000G  $SCSR4U::  .WORD  $SCSR4U
000770  000000G  $SCSR4V::  .WORD  $SCSR4V
000772  000000G  $SCSR4W::  .WORD  $SCSR4W
000774  000000G  $SCSR4X::  .WORD  $SCSR4X
000776  000000G  $SCSR4Y::  .WORD  $SCSR4Y
001000  000000G  $SCSR4Z::  .WORD  $SCSR4Z
001002  000000G  $SCSR50::  .WORD  $SCSR50
001004  000000G  $SCSR51::  .WORD  $SCSR51
001006  000000G  $SCSR52::  .WORD  $SCSR52
001010  000000G  $SCSR53::  .WORD  $SCSR53
001012  000000G  $SCSR54::  .WORD  $SCSR54
001014  000000G  $SCSR55::  .WORD  $SCSR55
001016  000000G  $SCSR56::  .WORD  $SCSR56
001020  000000G  $SCSR57::  .WORD  $SCSR57
001022  000000G  $SCSR58::  .WORD  $SCSR58
001024  000000G  $SCSR59::  .WORD  $SCSR59
001026  000000G  $SCSR5A::  .WORD  $SCSR5A
001030  000000G  $SCSR5B::  .WORD  $SCSR5B
001032  000000G  $SCSR5C::  .WORD  $SCSR5C
001034  000000G  $SCSR5D::  .WORD  $SCSR5D
001036  000000G  $SCSR5E::  .WORD  $SCSR5E
001040  000000G  $SCSR5F::  .WORD  $SCSR5F
001042  000000G  $SCSR5G::  .WORD  $SCSR5G
001044  000000G  $SCSR5H::  .WORD  $SCSR5H
001046  000000G  $SCSR5I::  .WORD  $SCSR5I
001050  000000G  $SCSR5J::  .WORD  $SCSR5J
001052  000000G  $SCSR5K::  .WORD  $SCSR5K
001054  000000G  $SCSR5L::  .WORD  $SCSR5L
001056  000000G  $SCSR5M::  .WORD  $SCSR5M
001060  000000G  $SCSR5N::  .WORD  $SCSR5N
001062  000000G  $SCSR5O::  .WORD  $SCSR5O
001064  000000G  $SCSR5P::  .WORD  $SCSR5P
001066  000000G  $SCSR5Q::  .WORD  $SCSR5Q
001070  000000G  $SCSR5R::  .WORD  $SCSR5R
001072  000000G  $SCSR5S::  .WORD  $SCSR5S
001074  000000G  $SCSR5T::  .WORD  $SCSR5T
001076  000000G  $SCSR5U::  .WORD  $SCSR5U
001100  000000G  $SCSR5V::  .WORD  $SCSR5V
001102  000000G  $SCSR5W::  .WORD  $SCSR5W
001104  000000G  $SCSR5X::  .WORD  $SCSR5X
001106  000000G  $SCSR5Y::  .WORD  $SCSR5Y
001110  000000G  $SCSR5Z::  .WORD  $SCSR5Z
001112  000000G  $SCSR60::  .WORD  $SCSR60
001114  000000G  $SCSR61::  .WORD  $SCSR61
001116  000000G  $SCSR62::  .WORD  $SCSR62
001120  000000G  $SCSR63::  .WORD  $SCSR63
001122  000000G  $SCSR64::  .WORD  $SCSR64
001124  000000G  $SCSR65::  .WORD  $SCSR65
001126  000000G  $SCSR66::  .WORD  $SCSR66
001130  000000G  $SCSR67::  .WORD  $SCSR67
001132  000000G  $SCSR68::  .WORD  $SCSR68
001134  000000G  $SCSR69::  .WORD  $SCSR69
001136  000000G  $SCSR6A::  .WORD  $SCSR6A
001140  000000G  $SCSR6B::  .WORD  $SCSR6B
001142  000000G  $SCSR6C::  .WORD  $SCSR6C
001144  000000G  $SCSR6D::  .WORD  $SCSR6D
001146  000000G  $SCSR6E::  .WORD  $SCSR6E
001150  000000G  $SCSR6F::  .WORD  $SCSR6F
001152  000000G  $SCSR6G::  .WORD  $SCSR6G
001154  000000G  $SCSR6H::  .WORD  $SCSR6H
001156  000000G  $SCSR6I::  .WORD  $SCSR6I
001160  000000G  $SCSR6J::  .WORD  $SCSR6J
001162  000000G  $SCSR6K::  .WORD  $SCSR6K
001164  000000G  $SCSR6L::  .WORD  $SCSR6L
001166  000000G  $SCSR6M::  .WORD  $SCSR6M
001170  000000G  $SCSR6N::  .WORD  $SCSR6N
001172  000000G  $SCSR6O::  .WORD  $SCSR6O
001174  000000G  $SCSR6P::  .WORD  $SCSR6P
001176  000000G  $SCSR6Q::  .WORD  $SCSR6Q
001200  000000G  $SCSR6R::  .WORD  $SCSR6R
001202  000000G  $SCSR6S::  .WORD  $SCSR6S
001204  000000G  $SCSR6T::  .WORD  $SCSR6T
001206  000000G  $SCSR6U::  .WORD  $SCSR6U
001210  000000G  $SCSR6V::  .WORD  $SCSR6V
001212  000000G  $SCSR6W::  .WORD  $SCSR6W
001214  000000G  $SCSR6X::  .WORD  $SCSR6X
001216  000000G  $SCSR6Y::  .WORD  $SCSR6Y
001220  000000G  $SCSR6Z::  .WORD  $SCSR6Z
001222  000000G  $SCSR70::  .WORD  $SCSR70
001224  000000G  $SCSR71::  .WORD  $SCSR71
001226  000000G  $SCSR72::  .WORD  $SCSR72
001230  000000G  $SCSR73::  .WORD  $SCSR73
001232  000000G  $SCSR74::  .WORD  $SCSR74
001234  000000G  $SCSR75::  .WORD  $SCSR75
001236  000000G  $SCSR76::  .WORD  $SCSR76
001240  000000G  $SCSR77::  .WORD  $SCSR77
001242  000000G  $SCSR78::  .WORD  $SCSR78
001244  000000G  $SCSR79::  .WORD  $SCSR79
001246  000000G  $SCSR7A::  .WORD  $SCSR7A
001250  000000G  $SCSR7B::  .WORD  $SCSR7B
001252  000000G  $SCSR7C::  .WORD  $SCSR7C
001254  000000G  $SCSR7D::  .WORD  $SCSR7D
001256  000000G  $SCSR7E::  .WORD  $SCSR7E
001260  000000G  $SCSR7F::  .WORD  $SCSR7F
001262  000000G  $SCSR7G::  .WORD  $SCSR7G
001264  000000G  $SCSR7H::  .WORD  $SCSR7H
001266  000000G  $SCSR7I::  .WORD  $SCSR7I
001270  000000G  $SCSR7J::  .WORD  $SCSR7J
001272  000000G  $SCSR7K::  .WORD  $SCSR7K
001274  000000G  $SCSR7L::  .WORD  $SCSR7L
001276  000000G  $SCSR7M::  .WORD  $SCSR7M
001300  000000G  $SCSR7N::  .WORD  $SCSR7N
001302  000000G  $SCSR7O::  .WORD  $SCSR7O
001304  000000G  $SCSR7P::  .WORD  $SCSR7P
001306  000000G  $SCSR7Q::  .WORD  $SCSR7Q
001310  000000G  $SCSR7R::  .WORD  $SCSR7R
001312  000000G  $SCSR7S::  .WORD  $SCSR7S
001314  000000G  $SCSR7T::  .WORD  $SCSR7T
001316  000000G  $SCSR7U::  .WORD  $SCSR7U
001320  000000G  $SCSR7V::  .WORD  $SCSR7V
001322  000000G  $SCSR7W::  .WORD  $SCSR7W
001324  000000G  $SCSR7X::  .WORD  $SCSR7X
001326  000000G  $SCSR7Y::  .WORD  $SCSR7Y
001330  000000G  $SCSR7Z::  .WORD  $SCSR7Z
001332  000000G  $SCSR80::  .WORD  $SCSR80
001334  000000G  $SCSR81::  .WORD  $SCSR81
001336  000000G  $SCSR82::  .WORD  $SCSR82
001340  000000G  $SCSR83::  .WORD  $SCSR83
001342  000000G  $SCSR84::  .WORD  $SCSR84
001344  000000G  $SCSR85::  .WORD  $SCSR85
001346  000000G  $SCSR86::  .WORD  $SCSR86
001350  000000G  $SCSR87::  .WORD  $SCSR87
001352  000000G  $SCSR88::  .WORD  $SCSR88
001354  000000G  $SCSR89::  .WORD  $SCSR89
001356  000000G  $SCSR8A::  .WORD  $SCSR8A
001360  000000G  $SCSR8B::  .WORD  $SCSR8B
001362  000000G  $SCSR8C::  .WORD  $SCSR8C
001364  000000G  $SCSR8D::  .WORD  $SCSR8D
001366  000000G  $SCSR8E::  .WORD  $SCSR8E
001370  000000G  $SCSR8F::  .WORD  $SCSR8F
001372  000000G  $SCSR8G::  .WORD  $SCSR8G
001374  000000G  $SCSR8H::  .WORD  $SCSR8H
001376  000000G  $SCSR8I::  .WORD  $SCSR8I
001400  000000G  $SCSR8J::  .WORD  $SCSR8J
001402  000000G  $SCSR8K::  .WORD  $SCSR8K
001404  000000G  $SCSR8L::  .WORD  $SCSR8L
001406  000000G  $SCSR8M::  .WORD  $SCSR8M
001410  000000G  $SCSR8N::  .WORD  $SCSR8N
001412  000000G  $SCSR8O::  .WORD  $SCSR8O
001414  000000G  $SCSR8P::  .WORD  $SCSR8P
001416  000000G  $SCSR8Q::  .WORD  $SCSR8Q
001420  000000G  $SCSR8R::  .WORD  $SCSR8R
001422  000000G  $SCSR8S::  .WORD  $SCSR8S
001424  000000G  $SCSR8T::  .WORD  $SCSR8T
001426  000000G  $SCSR8U::  .WORD  $SCSR8U
001430  000000G  $SCSR8V::  .WORD  $SCSR8V
001432  000000G  $SCSR8W::  .WORD  $SCSR8W
001434  000000G  $SCSR8X::  .WORD  $SCSR8X
001436  000000G  $SCSR8Y::  .WORD  $SCSR8Y
001440  000000G  $SCSR8Z::  .WORD  $SCSR8Z
001442  000000G  $SCSR90::  .WORD  $SCSR90
001444  000000G  $SCSR91::  .WORD  $SCSR91
001446  000000G  $SCSR92::  .WORD  $SCSR92
001450  000000G  $SCSR93::  .WORD  $SCSR93
001452  000000G  $SCSR94::  .WORD  $SCSR94
001454  000000G  $SCSR95::  .WORD  $SCSR95
001456  000000G  $SCSR96::  .WORD  $SCSR96
001460  000000G  $SCSR97::  .WORD  $SCSR97
001462  000000G  $SCSR98::  .WORD  $SCSR98
001464  000000G  $SCSR99::  .WORD  $SCSR99
001466  000000G  $SCSR9A::  .WORD  $SCSR9A
001470  000000G  $SCSR9B::  .WORD  $SCSR9B
001472  000000G  $SCSR9C::  .WORD  $SCSR9C
001474  000000G  $SCSR9D::  .WORD  $SCSR9D
001476  000000G  $SCSR9E::  .WORD  $SCSR9E
001500  000000G  $SCSR9F::  .WORD  $SCSR9F
001502  000000G  $SCSR9G::  .WORD  $SCSR9G
001504  000000G  $SCSR9H::  .WORD  $SCSR9H
001506  000000G  $SCSR9I::  .WORD  $SCSR9I
001510  000000G  $SCSR9J::  .WORD  $SCSR9J
001512  000000G  $SCSR9K::  .WORD  $SCSR9K
001514  000000G  $SCSR9L::  .WORD  $SCSR9L
001516  000000G  $SCSR9M::  .WORD  $SCSR9M
001520  000000G  $SCSR9N::  .WORD  $SCSR9N
001522  000000G  $SCSR9O::  .WORD  $SCSR9O
001524  000000G  $SCSR9P::  .WORD  $SCSR9P
001526  000000G  $SCSR9Q::  .WORD  $SCSR9Q
001530  000000G  $SCSR9R::  .WORD  $SCSR9R
001532  000000G  $SCSR9S::  .WORD  $SCSR9S
001534  000000G  $SCSR9T::  .WORD  $SCSR9T
001536  000000G  $SCSR9U::  .WORD  $SCSR9U
001540  000000G  $SCSR9V::  .WORD  $SCSR9V
001542  000000G  $SCSR9W::  .WORD  $SCSR9W
001544  000000G  $SCSR9X::  .WORD  $SCSR9X
001546  000000G  $SCSR9Y::  .WORD  $SCSR9Y
001550  000000G  $SCSR9Z::  .WORD  $SCSR9Z
001552  000000G  $SCSRA0::  .WORD  $SCSRA0
001554  000000G  $SCSRA1::  .WORD  $SCSRA1
001556  000000G  $SCSRA2::  .WORD  $SCSRA2
001560  000000G  $SCSRA3::  .WORD  $SCSRA3
001562  000000G  $SCSRA4::  .WORD  $SCSRA4
001564  000000G  $SCSRA5::  .WORD  $SCSRA5
001566  000000G  $SCSRA6::  .WORD  $SCSRA6
001570  000000G  $SCSRA7::  .WORD  $SCSRA7
001572  000000G  $SCSRA8::  .WORD  $SCSRA8
001574  000000G  $SCSRA9::  .WORD  $SCSRA9
001576  000000G  $SCSRAA::  .WORD  $SCSRAA
001600  000000G  $SCSRAB::  .WORD  $SCSRAB
001602  000000G  $SCSRAC::  .WORD  $SCSRAC
001604  000000G  $SCSRAD::  .WORD  $SCSRAD
001606  000000G  $SCSRAE::  .WORD  $SCSRAE
001610  000000G  $SCSRAF::  .WORD  $SCSRAF
001612  000000G  $SCSRAG::  .WORD  $SCSRAG
001614  000000G  $SCSRAH::  .WORD  $SCSRAH
001616  000000G  $SCSRAI::  .WORD  $SCSRAI
001620  000000G  $SCSRAJ::  .WORD  $SCSRAJ
001622  000000G  $SCSRAK::  .WORD  $SCSRAK
001624  000000G  $SCSRAL::  .WORD  $SCSRAL
001626  000000G  $SCSRAM::  .WORD  $SCSRAM
001630  000000G  $SCSRAN::  .WORD  $SCSRAN
001632  000000G  $SCSRAO::  .WORD  $SCSRAO
001634  000000G  $SCSRAP::  .WORD  $SCSRAP
001636  000000G  $SCSRAQ::  .WORD  $SCSRAQ
001640  000000G  $SCSRAR::  .WORD  $SCSRAR
001642  000000G  $SCSRAS::  .WORD  $SCSRAS
001644  000000G  $SCSRAT::  .WORD  $SCSRAT
001646  000000G  $SCSRAU::  .WORD  $SCSRAU
001650  000000G  $SCSRAV::  .WORD  $SCSRAV
001652  000000G  $SCSRAW::  .WORD  $SCSRAW
001654  000000G  $SCSRAX::  .WORD  $SCSRAX
001656  000000G  $SCSRAY::  .WORD  $SCSRAY
001660  000000G  $SCSRAZ::  .WORD  $SCSRAZ
001662  000000G  $SCSRB0::  .WORD  $SCSRB0
001664  000000G  $SCSRB1::  .WORD  $SCSRB1
001666  000000G  $SCSRB2::  .WORD  $SCSRB2
001670  000000G  $SCSRB3::  .WORD  $SCSRB3
001672  000000G  $SCSRB4::  .WORD  $SCSRB4
001674  000000G  $SCSRB5::  .WORD  $SCSRB5
001676  000000G  $SCSRB6::  .WORD  $SCSRB6
001700  000000G  $SCSRB7::  .WORD  $SCSRB7
001702  000000G  $SCSRB8::  .WORD  $SCSRB8
001704  000000G  $SCSRB9::  .WORD  $SCSRB9
001706  000000G  $SCSRBA::  .WORD  $SCSRBA
001710  000000G  $SCSRBB::  .WORD  $SCSRBB
001712  000000G  $SCSRBC::  .WORD  $SCSRBC
001714  000000G  $SCSRBD::  .WORD  $SCSRBD
001716  000000G  $SCSRBE::  .WORD  $SCSRBE
001720  000000G  $SCSRBF::  .WORD  $SCSRBF
001722  000000G  $SCSRBG::  .WORD  $SCSRBG
001724  000000G  $SCSRBH::  .WORD  $SCSRBH
001726  000000G  $SCSRBI::  .WORD  $SCSRBI
001730  000000G  $SCSRBJ::  .WORD  $SCSRBJ
001732  000000G  $SCSRBK::  .WORD  $SCSRBK
001734  000000G  $SCSRBL::  .WORD  $SCSRBL
001736  000000G  $SCSRBM::  .WORD  $SCSRBM
001740  000000G  $SCSRBN::  .WORD  $SCSRBN
001742  000000G  $SCSRBO::  .WORD  $SCSRBO
001744  000000G  $SCSRBP::  .WORD  $SCSRBP
001746  000000G  $SCSRBQ::  .WORD  $SCSRBQ
001750  000000G  $SCSRBR::  .WORD  $SCSRBR
001752  000000G  $SCSRBS::  .WORD  $SCSRBS
001754  000000G  $SCSRBT::  .WORD  $SCSRBT
001756  000000G  $SCSRBU::  .WORD  $SCSRBU
001760  000000G  $SCSRBV::  .WORD  $SCSRBV
001762  000000G  $SCSRBW::  .WORD  $SCSRBW
001764  000000G  $SCSRBX::  .WORD  $SCSRBX
001766  000000G  $SCSRBY::  .WORD  $SCSRBY
001770  000000G  $SCSRBZ::  .WORD  $SCSRBZ
001772  000000G  $SCSRC0::  .WORD  $SCSRC0
001774  000000G  $SCSRC1::  .WORD  $SCSRC1
001776  000000G  $SCSRC2::  .WORD  $SCSRC2
002000  000000G  $SCSRC3::  .WORD  $SCSRC3
002002  000000G  $SCSRC4::  .WORD  $SCSRC4
002004  000000G  $SCSRC5::  .WORD  $SCSRC5
002006  000000G  $SCSRC6::  .WORD  $SCSRC6
002010  000000G  $SCSRC7::  .WORD  $SCSRC7
002012  000000G  $SCSRC8::  .WORD  $SCSRC8
002014  000000G  $SCSRC9::  .WORD  $SCSRC9
002016  000000G  $SCSRCA::  .WORD  $SCSRCA
002020  000000G  $SCSRCB::  .WORD  $SCSRCB
002022  000000G  $SCSRCC::  .WORD  $SCSRCC
002024  000000G  $SCSRC
```

```
                                        .SBTTL  $SCEX - SET CEX

                                ;+
                                ; $SCEX - "SET CEX"
                                ;
                                ; INPUTS:
                                ;       FL.NTI OF $FLAGS SET IF NTINIT WORK IS TO BE DONE
                                ;       ELSE .HOME = HOME BLOCK BIAS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;
                                ;-
000552                          $SCEX:
000552  004767  000000G                CALL    $NTICF          ; SCAN CONFIG FILE FOR NTPOOL PARAMETERS
000556  103423                        BCS     60$             ; IF CS, ERROR
000560  004767  000000G                CALL    S$CEX           ; TRANSFER COMM EXEC INTO CORE
000564  103420                        BCS     60$             ; IF CS, ERROR
000566  004767  000000G                CALL    $NTICF          ; SCAN CONFIG FILE FOR MARKED-FOR-LOAD
000572  103415                        BCS     60$             ; IF CS, ERROR

                                        .IF NDF I$$AS

000574  005727  000000G                TST     #$NTCF3         ; PSI SUPPORTED?
000600  001403                        BEQ     10$             ; -NO
000602  004767  000000G                CALL    $NTCF3          ; SCAN CONFIG FILE FOR PSI COMPONENTS
000606  103407                        BCS     60$             ; IF CS, ERROR
000610                          10$:
000610  005727  000000G                TST     #$NTCFS         ; SNA GATEWAY SUPPORTED?
000614  001403                        BEQ     20$             ; -NO
000616  004767  000000G                CALL    $NTCFS          ; SCAN CONFIG FILE FOR SNA COMPONENTS
000622  103401                        BCS     60$             ; IF CS, ERROR
000624                          20$:
                                        .ENDC

000624  000403                        BR      70$             ; RETURN, ALL DONE

000626  004767  000000G        60$:    CALL    $CCEX           ; CLEANUP AFTER THE ERROR
000632  000261                        SEC                     ; INDICATE FAILURE
000634  000207                70$:    RETURN
```

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 9

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
.UMRFL	= 000516 RG	#6-372 6-373
..SCSR	000060 RG	#6-285

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 10

```
                                        .SBTTL  PDVA - FIND PDV ADDRESS

                                ;+
                                ; PDVA - FIND PDV ADDRESS
                                ;
                                ; INPUTS:
                                ;       R3=REQUEST BLOCK POINTER
                                ;               .BLKW   1       ; LR.PRO - PROCESS NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       $PDVA=PDV ADDRESS
                                ;       $NAME=RAD50 PDV NAME
                                ;       $FLAGS SET TO PROCESS TYPE FROM PDV STATUS BYTE
                                ;
                                ;-

000204  012300                 PDVA:   MOV     (R3)+,R0        ; GET PROCESS NAME
000206  001417                         BEQ     101$            ; IF ZERO, RETURN FAILURE
000210  010067  000000G                MOV     R0,$NAME        ; STORE IT
000214                                  CALL    $FPDV           ; TRY TO FIND PDV
000220  103412                         BCS     101$            ; IF CS, NOT FOUND
000222  010067  000000G                MOV     R0,$PDVA        ; SET PDV ADDRESS
000226  116001  000000G                MOVB    Z.FLG(R0),R1    ; GET PROCESS TYPE FROM PDV STATUS BYTE
000232  042701  000000C                BIC     #^C<FL.DDM!FL.DLC!FL.LLC>,R1    ; ...
000236  001406                         BEQ     111$            ; IF ZERO, IT'S NOT A REAL PDV!
000240  050167  000000G                BIS     R1,$FLAGS       ; UPDATE NETLDR STATUS
000244                                  RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000246                          101$:   EMSG$R  97              ; PROCESS NOT IN SYSTEM
000254                          111$:   EMSG$R  96              ; GARBAGE IN PDV
```

```
                                .SBTTL  KMCPF - CHECK KMC POWERFAIL RELOAD REQUEST BLOCK

                        ;+
                        ;
                        ; KMCPF - CHECK KMC POWERFAIL RELOAD REQUEST BLOCK
                        ;
                        ; INPUTS:
                        ;       NONE
                        ;
                        ; OUTPUTS:
                        ;       R4 CLEARED IF NO WORK TO DO
                        ;
                        ;-

002120                  KMCPF:
                                .IF DF  M$$MGE

002120  017700  000000G         MOV     @NTLPT,R0       ; GET NETLDR DATA BLOCK BIAS

                                .IF DF  R$$11M

002124  010403                  MOV     R4,R3           ; MAKE A COPY OF THE FUNCTION CODE
002126                          SWSTK$  20$             ;* ENTER KERNEL MODE

                                .IFTF

002132                          CALL    $MAPX           ;* MAP TO THE DATA BLOCK
002136  005762  000000G         TST     .CXCSR(R2)      ;* ARE THERE ANY KMC'S LOADED?
002142  001002                  BNE     10$             ;* IF NE, YES

                                .IFT

002144  005066  000010          CLR     10(SP)          ;* ZERO USER MODE R3
002150                  10$:    RETURN                  ;* BACK TO USER MODE
002152  010304          20$:    MOV     R3,R4           ; POSSIBLE CLEAR OF FUNCTION CODE
002154                          RETURN

                                .IFF

                                CLR     R4              ;; INDICATE NO WORK TO DO
                        10$:    RETURN                  ;; RE-ENABLE TASK SWITCHING AND EXIT

                                .ENDC

                                .IFF

                                MOV     @NTLPT,R2       ; GET NETLDR DATA BLOCK ADDRESS
                                TST     .CXCSR(R2)      ; ARE ANY KMC'S LOADED?
                                BNE     10$             ; IF NE, YES
                                CLR     R4              ; INDICATE NO WORK TO DO
                        10$:    RETURN

                                .ENDC
```

```
                                        .SBTTL  REP7B - REPLACE REQUESTING TASK'S NAME AND LINE ID

                                ;+
                                ; REP7B - REPLACE REQUESTING TASK'S NAME AND LINE ID
                                ;
                                ; INPUTS:
                                ;       $$TASK=REQUESTING TASK'S NAME
                                ;       R2=POINTER INTO SLT MAPPING TABLE + 2
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       R3,R4=UPDATED
                                ;
                                ;-
003564  016246  177776          REP7B:  MOV     -2(R2),-(SP)    ; SAVE SLT ADDRESS
003570                                  CALL    REP7            ; REPLACE REQUESTING TASK'S NAME
003574  112314                  10$:    MOVB    (R3)+,(R4)      ; LOOK FOR THE SECOND ASTERISK
003576  122724  000052                  CMPB    #'*,(R4)+       ;       ...
003602  001374                          BNE     10$             ;       ...
003604  005304                          DEC     R4              ; SKIP THE ASTERISK
003606  010400                          MOV     R4,R0           ; COPY BUFFER ADDRESS
003610  011601                          MOV     (SP),R1         ; GET SLT ADDRESS
003612  116101  000002                  MOVB    L.DDM(R1),R1    ; AND DDM PDV INDEX
003616  067701  000000G                 ADD     @PDVTA,R1       ; ADD PDV TABLE ADDRESS
003622  016101  000000G                 MOV     .CBIAS(R1),R1   ; GET PDV ADDRESS

003626  016101  000000G         15$:    MOV     Z.NAM(R1),R1    ; GET DDM PROCESS NAME
003632                                  CALL    $C5TA           ; CONVERT TO ASCII
003636  122740  000040          20$:    CMPB    #' ,-(R0)       ; DISCARD TRAILING BLANKS
003642  001775                          BEQ     20$             ;       ...
003644  005200                          INC     R0              ;       ...
003646  112720  000055                  MOVB    #'-,(R0)+       ; DEPOSIT A HYPHEN
003652  112720  000060                  MOVB    #'0,(R0)+       ; AND AN ASCII ZERO
003656  011601                          MOV     (SP),R1         ; GET SLT ADDRESS
003660  116101  000012                  MOVB    L.CTL(R1),R1    ; AND CONTROLLER NUMBER
003664  001404                          BEQ     30$             ; IF ZERO, NO WORK TO DO
003666  005300                          DEC     R0              ; SKIP THE ASCII ZERO
003670  005002                          CLR     R2              ; SUPPRESS LEADING ZEROES
003672                                  CALL    $CBOMG          ; CONVERT TO OCTAL
003676  012601                  30$:    MOV     (SP)+,R1        ; GET SLT ADDRESS
003700  116101  000013                  MOVB    L.UNT(R1),R1    ; AND UNIT NUMBER
003704  001405                          BEQ     40$             ; IF ZERO, NO WORK TO DO
003706  112720  000055                  MOVB    #'-,(R0)+       ; DEPOSIT ANOTHER HYPHEN
003712  005002                          CLR     R2              ; SUPPRESS LEADING ZEROES
003714                                  CALL    $CBOMG          ; CONVERT TO OCTAL
003720  010004                  40$:    MOV     R0,R4           ; SET NEW BUFFER ADDRESS
003722                                  RETURN

000001                                  .END
```

```
                                        $CFUIC::.IF DF  R$$11M
000052  123  131  072                            .ASCIZ  "SY:[130,10]"
000066                                          .BLKB   4
                                                .IFF
                                                .ASCIZ  "SY:[11,61]"
                                                .BLKB   5
                                                .ENDC

                                        ;
                                        ; COMM EXEC UIC
                                        ;
                                                .IF DF  R$$11M
000072                                  $CXUIC::.IF DF  M$$MGE
000072  123  131  072                            .ASCIZ  "SY:[130,54]"
                                                .IFF
                                                .ASCIZ  "SY:[130,50]"
                                                .ENDC
000106                                          .BLKB   4
                                                .ENDC

                                        ;
                                        ; DEFAULT FILE NAME
                                        ;
000112  130  130  130  $DFNAM::.ASCIZ  "XXXXXX"

                                        ;
                                        ; DEFAULT EXTENSION
                                        ;
000121  124  123  113  DFEXT:  .ASCIZ  "TSK"

                                        ;
                                        ; FILE NAME(S) STORAGE AREA
                                        ;
000125                  NAMES:  .BLKB   7*10.

                                        ;
                                        ; GENERAL PURPOSE FILE NAME (READY FOR OPEN FAILURE SECOND ERROR MESSAGE LINE)
                                        ;
000233  015            FNERR:  .BYTE   15
000234  116  124  114           .ASCII  "NTL -- "
000243  123  131  060  FILE:   .ASCIZ  "SY00:[XXX,XXX]XXXXXXXXX.XXX"

                                        .EVEN

000000                                  .PSECT
```

```
                        .TITLE  NTLSYM - OFFSET SYMBOL DEFINITIONS
                        .IDENT  /V05.00/

;
;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - GLOBAL SYMBOL DEFINITIONS
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                $ADDR:: .BLKW   1               ; ADDRESS
                                $MXADD::.BLKW   1               ; ADDRESS OF MUX TABLES
                                $MXSIZ::.BLKW   1               ; SIZE OF ALL MUX TABLES (ONLY FOR .MXPTB)
                                $OFF::  .BLKW   1               ; OFFSET VALUE/START OF MUX TABLES
                                $MXLEN::.BLKW   1               ; MUX TABLE LENGTH (KEPT AS NEGATIVE)
                                $NALL:: .BLKB   1               ; NUMBER OF ALLOCATIONS
                                $NLIB:: .BLKB   1               ; NUMBER OF LIBRARY REFERENCES
                                $SZPTR::.BLKW   1               ; POINTER INTO SIZE TABLE
                                        .BLKW   6               ; $SIZE SAVED EACH TIME A NEW LEVEL IS ENTERED
                                        .IF DF  M$$MGE
                                $UBIAS::.BLKW   1               ; UNLOAD BLOCK BIAS
                                        .ENDC
                                $UDA::  .BLKW   1               ; UNLOAD DATA VIRTUAL ADDRESS
                                $LDA::  .BLKW   1               ; LIBRARY DATA VIRTUAL ADDRESS
                                $MPLD:: .BLKW   1               ; ".MPLHD" DISPLACEMENT
                                $LFLD:: .BLKW   1               ; ".LFLHD" DISPLACEMENT

                                ;
                                ; DDM VECTOR INFORMATION
                                ;
                                $VECT1::.BLKW   1               ; DISPLACEMENT OF ".INT1" FIELD
                                $VECT2::.BLKW   1               ; DITTO ".INT2"
                                $VECT3::.BLKW   1               ; DITTO ".INT3"
                                $NVECT::.BLKW   1               ; NUMBER OF VECTORS
                                $ICBAD::.BLKW   1               ; ADDRESS OF ICB
                                $ICBSZ::.BLKW   1               ; SIZE OF ICB
        000012                  $JSROF  == 12                   ; OFFSET FROM $VECTX TO "JSR @#..."
        120000                  $PRAPR  == 120000               ; ADDRESS TO USE FOR PARTITION LINE TABLES
                                                                ;     USE KERNEL APR5

                                ;
                                ; TPARS ACTION ROUTINES INFORMATION
                                ;
                                $VALUE::.BLKW   1               ; EXPRESSION VALUE
                                $FILEN::.BLKB   10.             ; ASCIZ FILE NAME
                                $DIFF:: .BLKW   1               ; MODIFICATION VALUE
                                $BFALL::.BLKW   1               ; ".BIN" ALLOCATION SIZE IN 32. WORD BLOCKS
                                $BFLBN::.BLKW   2               ; ".BIN" LOGICAL BLOCK NUMBER
                                $BFPTR::.BLKW   1               ; ".BIN" BINARY BUFFER POINTER
                                $BFXFR::.BLKW   1               ; ".BIN" TRANSFER SIZE IN BYTES
                                $BFBAS::.BLKW   1               ; BINARY/LIBRARY FILE TKB BASE ADDRESS
                                $FNR50::.BLKW   3               ; FILE NAME IN RAD50
                                $CVAL:: .BLKW   1               ; ".CORE" OPERAND 1 VALUE
                                $LBADR::.BLKW   1               ; ".LIBR" DESCRIPTOR BLOCK ADDRESS
                                        .IF DF  M$$MGE
                                $CAPR:: .BLKW   1               ; ".CORE" OPERAND 2 APR NUMBER
                                $LBIAS::.BLKW   1               ; ".LIBR" DESCRIPTOR BLOCK BIAS
                                        .ENDC

                                ;
                                ; TEMPLATE BUFFER AND BUFFER SIZE
                                ;
                                TMPBF:  .BLKB   132.
                                TMPSZ:  .BLKW   1

                                        .NLIST  BEX
```

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES								
EPFX		000507	R	#5-180	15-488							
ERRBUF		000361	R	#5-174	15-491							
ERRMSG		000354	R	#5-172	15-506							
ERR$10	=	******	GX	7-294								
ERR$11	=	******	GX	7-295								
ERR$12	=	******	GX	7-296								
ERR$9	=	******	GX	6-257								
FL.DDM	=	******	GX	12-413								
FL.DLC	=	******	GX	13-428								
FL.ECH	=	******	GX	6-249								
FL.LLC	=	******	GX	15-484								
FMT2		000740	R	5-189	5-190	5-191	5-192	5-193	#5-202			
FM.2	=	000000		#5-189	#5-190	#5-191	#5-192	#5-193				
F1.BBO	=	000001	G	#5-71	10-367	10-372						
F1.ERR	=	100000	G	#5-85	15-487							
F1.INT	=	000002	G	#5-72								
F1.IN1	=	000004	G	#5-73								
F1.IN2	=	000010	G	#5-74								
F1.IN3	=	000020	G	#5-75								
F1.LFI	=	000040	G	#5-76								
F1.LFL	=	000100	G	#5-77								
F1.MPL	=	000200	G	#5-78								
F1.MPT	=	000400	G	#5-79								
F1.MXT	=	001000	G	#5-80								
F1.STE	=	002000	G	#5-81								
F1.UMR	=	004000	G	#5-82								
F1.VCT	=	020000	G	#5-84								
F1.VSE	=	010000	G	#5-83								
F2.BYT	=	000001	G	#5-93								
F2.END	=	000002	G	#5-94	6-258							
F2.FIL	=	000004	G	#5-95								
F2.LBR	=	000010	G	#5-96								
F2.NEG	=	000020	G	#5-97								
F2.SUB	=	000040	G	#5-98								
IE.EOF	=	******	GX	7-290								
IE.RBG	=	******	GX	7-292								
M$$MGE	=	000000		5-121	5-156							
S$$BAS	=	******		5-189	5-189	5-190	5-190	5-191	5-191	5-192	5-192	5-193
				5-193								
TMPBF		000146	R	#5-164	7-278	7-282	9-329					
TMPSZ		000352	R	#5-165	*7-281	9-330						
TM.RD		000210	R	6-247	#7-278							
WPFX		000172	R	#5-179	15-479							
$ABFI	=	******	GX	8-313								
$ADDR		000016	RG	#5-112								
$BFALL		000114	RG	#5-148								
$BFBAS		000126	RG	#5-152								
$BFLBN		000116	RG	#5-149								
$BFPTR		000122	RG	#5-150								
$BFXFR		000124	RG	#5-151								
$BIN		000004	RG	#5-103								
$CAPR		000142	RG	#5-157								

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 2

NTLDEV CREATED BY MACRO ON 29-JUN-85 AT 00:30 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	8-243	8-245	8-249	8-251	8-253	8-255	8-272	9-320	10-370	10-375
	10-387	10-389	10-396	10-404	10-408	11-439	11-440	11-446	12-487	12-488
	13-539	13-553	13-556	13-561	13-564	13-581	13-586	16-699	16-703	16-716
	16-719	17-741	17-742	17-747	18-765	18-774	18-787	18-789		
CALLR	10-417	17-745	17-748							
EMSG$	#5-59	10-396	10-404	10-408						
EMSG$R	#5-59	8-279	9-345	9-346	11-460	11-461	11-463	11-464	12-513	13-611
	13-612	13-613	13-616	13-617	13-618	13-621	13-622			
NTLER$	#5-59	7-156	7-157	7-158	7-159	7-160	7-161	7-162	7-163	7-164
	7-165	7-166	7-167	7-168	7-169	7-170	7-171	7-172	7-173	7-174
	7-175	7-176	7-177							
RESRG	#5-59									
RETURN	8-273	9-327	9-340	10-411	11-455	12-491	12-508	13-600	14-641	16-721
	18-795	19-812								
SAVRG	#5-59									
SWSTK$	9-320	12-487	13-586	17-742						

FILEID**NTLGET

NTLGET

MACRO CR

MACRO NA

CALL
DIR$
EMSG$
EXIT$S
NTLDF$
NTLER$
OFF$
QIOW$
RCVX$
RETC
RETURN
RSXOK$
R50$
SDAT$
SWSTK$

FILE

NTLGET CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES								
CALL	6-130	6-140	6-150	6-152	6-153	8-197	9-228	9-230	9-234	9-249
DIR$	#5-58	7-168	8-208							
EMSG$	#5-58	6-150								
EXIT$S	#5-58									
NTLDF$	#5-58	5-60								
NTLER$	#5-58	5-99								
OFF$	#5-81	#5-86								
QIOW$	#5-58									
RCVX$	#5-58	5-81								
RETC	#5-58	9-236								
RETURN	6-145	7-180	8-209	9-250						
RSXOK$	#5-68	6-126								
R50$	#5-81	5-81	#5-86	5-86						
SDAT$	#5-58	5-86								
SWSTK$	9-228									

NTLKMC
KMC LOA

FILEID**NTLKMC

NTLKMC
KMC LOA

```
                                        .SBTTL  KMC LOAD LIST HOUSEKEEPER

                                ;+
                                ; -KMCLL- KMC LOAD LIST HOUSEKEEPER
                                ;
                                ; THIS ROUTINE WILL KEEP HOUSE FOR THE KMC LOAD LIST.  THIS IS A
                                ; LIST OF KMC'S WHICH HAVE ALREADY HAD THEIR MICRO-CODE LOADED
                                ; (OR AT LEAST HAVE BEEN TRIED).  IT HOLDS ALL THE INFORMATION
                                ; NECESSARY TO RELOAD THE MICRO-CODE AFTER A POWER-FAIL.
                                ;
                                ; INPUTS:
                                ;       $NAME  = RAD50 PDV/DEVICE NAME
                                ;       $SLTA  = SYSTEM LINE TABLE ENTRY ADDRESS
                                ;       $$CSR  = KMC CSR ADDRESS
                                ;       $NTLPT = ADDRESS OF NTL CONTROL DATA BLOCK
                                ;       .CXKMC IN DATA BLOCK CONTAINS ADDRESS OF FIRST LOAD LIST BLOCK
                                ;         (CONTAINS ZERO IF NO BLOCKS ALLOCATED YET)
                                ;       NO REGISTER SAVING IS NECESSARY
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;-
000234                          KMCLL:
000234  005067  000014'         KMCLL:  CLR     KMCPN1                  ;B-INIT "LINK TO NEW BLOCK" POINTER
000240  016767  000000G 000000' KMCLL:  MOV     $NAME,KMCNAM            ;B-COPY STUFF BECAUSE WILL LOSE
000246  016700  000000G         KMCLL:  MOV     $SLTA,R0                ;B-  MAPPING TO ROOT SEGMENT
000252  116067  000012  000004' KMCLL:  MOVB    L.CTL(R0),KMCNUM        ;B-GET THE CONTROLLER NAME AND NUMBER
000260  105067  000005'         KMCLL:  CLRB    KMCNUM+1                ;B-ZERO LOCAL MICRO-CODE LENGTH
000264  016767  000000G 000002' KMCLL:  MOV     $$CSR,KMCCSR            ;B-GET THE KMC CSR ADDRESS

                                KMCLL:  .IF DF  M$$MGE
                                KMCLL:    .IF   DF,R$$11M
000272                          KMCLL:  SWSTK$  55$                     ;M-SWITCH STACKS, GO TO 55$ ON EXIT
                                KMCLL:    .ENDC
000276  016700  000000G         KMCLL:  MOV     $NTLPT,R0               ;M-GET BIAS OF THE DATA BLOCK
000302                          KMCLL:  CALL    $MAPX                   ;M-MAP IT, SET R1=MAP REGISTER, AND
000306  010203                  KMCLL:  MOV     R2,R3                   ;M- R2 (AND R3) -> BASE OF BLOCK
                                KMCLL:  .IFF
                                KMCLL:  MOV     $NTLPT,R3               ;U-POINT AT THE DATA BLOCK
                                KMCLL:  .ENDC

000310  062703  000000G         KMCLL:  ADD     #.CXKMC,R3              ;B-POINT AT FIRST LOAD LIST LINK
                                ;
                                ; GET THE ADDRESS OF THE NEXT LOAD LIST BLOCK
                                ;
000314  011300                  10$:    MOV     (R3),R0                 ;B-COPY ADDRESS OF LIST FROM LINK
000316  001427                  10$:    BEQ     45$                     ;B-BR IF NO NEXT LIST

                                10$:    .IF DF  M$$MGE
000320  010011                  10$:    MOV     R0,(R1)                 ;M-MAP TO LIST
000322  010200                  10$:    MOV     R2,R0                   ;M-POINT R0 AT LIST
000324  010203                  10$:    MOV     R2,R3                   ;M-RESET R3 TO BEGINNING OF BLOCK
                                10$:    .IFF
                                10$:    MOV     R0,R3                   ;U-SET TO POINT AT NEXT LINK
                                10$:    .ENDC

000326  005720                  10$:    TST     (R0)+                   ;B-POINT AT FIRST ENTRY
```

NTLKMC - LOAD KMC MICRO-CODE MACRO V05.03b Saturday 29-Jun-85 00:31 Page 22-1
Symbol table

```
A$$CHK= 000000          KMCFEO  001320R        LN.DOP= 000004          M$$CVR= 000000          X$$DBT= 000000
A$$CPS= 000000          KMCLD   000542R        LN.OPE= 000001          N$$ACC= 000001          ZF.DIA= ****** GX
A$$PRI= 000000          KMCLL   000234R        LN.REF= 000002          N$$BUF= 000001          Z.FLG = ****** GX
A$$TRP= 000000          KMCMP1  000012R    002 LN.SER= 000002          N$$LDV= 000001          $ALN18= ****** GX
C$$CKP= 000000          KMCNAM  000000R    002 LN.STA= 000017          N$$MCP= 000001          $AZFS = ****** GX
C$$ORE= 000400          KMCNUM  000004R    002 LN.SUB= 000360          N$$MLL= 000001          $CBDSG= ****** GX
C$$RSH= 177564          KMCPN1  000014R    002 LN.TRI= 000006          N$$MOV= 000010          $CBOMG= ****** GX
D$$BUG= 177514          KMCRL1  001372R        LOAD    000624R         N$$NCT= 000001          $CLDEQ= ****** GX
D$$ISK= 000000          KMCRL2  001444R        L$BFLG= ****** GX       N$$PEM= 000001          $CLIOS= ****** GX
D$$L11= 000001          KM.ALL= 000001         L$BLDZ= ****** GX       P$$P45= 000000          $CLOPE= ****** GX
D$$YNC= 000000          K$$CNT= 177546         L$BMXZ= ****** GX       P$$WRD= 000000          $C5TA = ****** GX
D$$YNM= 000000          K$$CSR= 177546         L$$ASG= 000000          Q$$OPT= 000010          $DFN2 = ****** GX
E$$XPR= 000000          K$$LDC= 000000         L$$DRV= 000000          R$$DER= 000000          $ERR3A  000142R     002
FMT10A  000512R     002 K$$TPS= 000074         L$$P11= 000001          R$$K11= 000001          $ERR3C  000174R     002
FMT10B  000540R     002 LD$LP = 000000         L$$11R= 000000          R$$SND= 000000          $ERR3D  000230R     002
FMT10C  000615R     002 LF.ACT= 100000         L.COST  000015          R$$11M= 000000          $ERR3E  000276R     002
FM.10A= 000000          LF.BRO= 000400         L.CTL   000012          SF.ACT= 000200          $ERR3F  000342R     002
FM.10B= 000000          LF.BWT= 000007         L.CVA   177776          SF.ENA= 000100          $ERR3H  000404R     002
FM.10C= 000000          LF.ENA= 002000         L.DDM   000002          SF.LPB= 000004          $ERR3I  000436R     002
F$$LVL= 000001          LF.LPB= 001000         L.DDS   000004          SF.MFL= 000040          $ERR3J  000500R     002
G$$TPP= 000000          LF.MDC= 000100         L.DLC   000003          SF.PAC= 000020          $IOSB = ****** GX
G$$TSS= 000000          LF.MFL= 004000         L.DLM   000006          SF.REA= 000010          $KMC    000000RG
G$$TTK= 000000          LF.MTP= 000020         L.DLS   000010          SF.SER= 000001          $KMCPF  000036RG
G$$WRD= 000000          LF.PAC= 000200         L.FLG   000000          SF.SVC= 000002          $LBN  = ****** GX
I$$RAR= 000000          LF.RDY= 040000         L.KRBA  000016          SF.UNL= 000040          $MAPX = ****** GX
I$$RDN= 000000          LF.REA= 010000         L.LEN = 000022          S$$WRG= 000000          $NAME = ****** GX
KMCBUF  000000R     003 LF.SER= 000040         L.MPr   000022          S$$YSZ= 007600          $NTLPT= ****** GX
KMCCSR  000002R     002 LF.TIM= 000010         L.NMST  000020          S.COST  000001          $PDVA = ****** GX
KMCDEV  000006R     002 LF.UNL= 020000         L.NSTA  000014          S.FLG   000000          $READ = ****** GX
KMCEBF  000024R     002 LF.X2P= 000000         L.OWNR  000021          S.LEN   000004          $REP.C= ****** GX
KMCEFT  001234R         LN.CLO= 000000         L.UNT   000013          S.NMST  000002          $REP.P= ****** GX
KMCEMS  000034R     002 LN.DUM= 000005         M$$CRB= 000124          S.OWNR  000003          $RLBL = ****** GX
KMCER1  000016R     002 LN.LOA= 000004         M$$CRX= 000000          TS$NHD= ****** GX       $SLTA = ****** GX
KMCER2  000020R     002 LN.LOO= 000003         M$$FCS= 000000          T$$KMG= 000000          $TIOUT= ****** GX
KMCER3  000022R     002 LN.OAU= 000003         M$$MGE= 000000          T$$MIN= 000000          $$CSR = ****** GX
KMCEX   001232R         LN.OFF= 000001         M$$NET= 000000          V$$CTR= 001000          .CXKMC= ****** GX
KMCFEC  001312R         LN.ON = 000000

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001532    001   (RW,I,LCL,REL,CON)
DATA    000700    002   (RW,D,LCL,REL,CON)
..KMC   000002    003   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12053 Words  ( 48 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:26.16
SY:NTLKMC.V2,[132,134]NTLKMC/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLKMC
```


NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 1

```
000212  103402                          BCS     111$            ; IF CS, SYMBOLS MISSING
000214                          20$:    RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000216                          101$:   CRASH                   ; $XFER SHOULD NOT HAVE BEEN CALLED
000220                          111$:   EMSG$R  QX              ; AUX TIMER SYMBOLS MISSING
000226  016000  000172'         TSKERR: MOV     ETAB2-2(R0),R0  ; GET ERROR MESSAGE CONTROL BLOCK ADDRESS
000232                                  CALLR   @(R0)+          ; PRINT ERROR MESSAGE
```

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
AXSYM	000234 R	7-312 #8-340
CMFRK	= ****** GX	8-344 8-356
DECPT	= ****** GX	12-630 12-646 13-684
DSPTAB	000012 R	#6-118 10-465 10-477 *10-519 *10-520 10-521
DTBLK	000056 R	6-186 #6-193
DTSYM	000062 R	#6-194 *10-524
DTVAL	000066 R	#6-195 10-542
D$MNOD	000041	13-686
D$NA	000062	12-633 12-656
D$NN	000060	12-631 12-647
ETAB1	000164 R	#6-228 10-583
ETAB2	000174 R	#6-236 7-322
E$NBR	000014	#5-62
E$NBS	000020	#5-62
E$NCR	000034	#5-62
E$NCS	000036	#5-62
E$NIC	000044	#5-62
E$NLEN	000050	#5-62 13-688 13-711 13-715 13-721
E$NLLA	000012	#5-62
E$NLNK	000000	#5-62
E$NML	000040	#5-62
E$NMR	000024	#5-62
E$NMS	000030	#5-62
E$NNOD	000002	#5-62
E$NRT	000042	#5-62
E$NRTP	000005	#5-62
E$NSEG	000010	#5-62
E$NTIM	000046	#5-62
E$NUSE	000004	#5-62
E$STRT	000006	#5-62
I$$AS	= ******	7-297 7-301 12-641 13-693
LF.ACT	= 100000	#5-65
LF.BRO	= 000400	#5-65
LF.BWT	= 000007	#5-65
LF.ENA	= 002000	#5-65
LF.LPB	= 001000	#5-65
LF.MDC	= 000100	#5-65
LF.MFL	= 004000	#5-65
LF.MTP	= 000020	#5-65
LF.PAC	= 000200	#5-65
LF.RDY	= 040000	#5-65
LF.REA	= 010000	#5-65
LF.SER	= 000040	#5-65
LF.TIM	= 000010	#5-65
LF.UNL	= 020000	#5-65
LF.X2P	= 000000	#5-65
LN.CLO	= 000000	#5-65
LN.DUM	= 000005	#5-65
LN.LOA	= 000004	#5-65
LN.LOO	= 000003	#5-65
LN.OAU	= 000003	#5-65
LN.OFF	= 000001	#5-65

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 2

```
                                ;
                                ; GENERAL INPUT PARAMETERS
                                ;
000000                          $RSXNM::.BLKW   1               ; ADDRESS OF $SYSNM
000002                          $$AST:: .BLKW   1               ; NOTIFICATION AST ADDRESS
000004                          $$PRM:: .BLKW   1               ; NOTIFICATION AST PARAMETER TO RETURN
000006                          $$TASK::.BLKW   2               ; REQUESTING TASK'S NAME
000012                          $FUNC:: .BLKW   1               ; FUNCTION CODE
000014                          $MISS:: .BLKW   1               ; MISSING PARAMETERS BIT MASK

                                        .IF DF  R$$MPL
                                        .IFTF
000016                          $ICBI   == 16                   ; ICB LENGTH I-SPACE ONLY
                                        .IFT
                                $ICBD   == 30                   ; ICB LENGTH I/D-SPACE SYSTEMS
                                $VCTB:: .WORD   0               ; ADDRESS OF PROCESS VECTOR TABLE
                                $VCLN:: .WORD   0               ; SIZE OF PROCESS VECTOR TABLE (WORDS)
                                $LIBNM::.BLKB   10.             ; ASCII NAME OF LIBRARY
                                $LBVL:: .BLKW   1               ; VALUE OF $XXXVL FOR LIBRARY
                                $LBVT:: .BLKW   1               ; VALUE OF $XXXVT FOR LIBRARY
                                $LBBLK::.BLKW   1               ; VALUE OF LIBRARY BIAS

                                        .IFTF
000016  000016                  $ICBLN::.WORD   $ICBI           ; DEFAULT ICB LENGTH TO I-SPACE ONLY

                                        .ENDC


000001                          MS.CSR==1                       ; CSR MISSING
000002                          MS.PRI==2                       ; INTERRUPT PRIORITY MISSING
000004                          MS.VCT==4                       ; VECTOR MISSING
000010                          MS.MTP==10                      ; MULTIPOINT STATION NUMBERS MISSING
004020                          MS.MUX==4020                    ; MUX UNIT INFORMATION MISSING
004037                          MS.LIN==4037
000040                          MS.LLC==40                      ; LLC INFO MISSING
000100                          MS.BUF==100                     ; BUF$DF MISSING
000200                          MS.PAR==200                     ; PAR$DF MISSING
000300                          MS.NT1==300                     ; NTINIT PHASE 1 INFO MISSING
000400                          MS.NT2==400                     ; NTINIT PHASE 2 INFO MISSING
001000                          MS.INC==1000                    ; LINE TABLE COUNT MISSING
002000                          MS.EXT==2000                    ; CONTROLLER TABLE COUNT MISSING
004000                          MS.SEC==4000                    ; SECONDARY CSR MISSING
010000                          MS.CIR==10000                   ; CIRCUIT PARAMETERS ARE MISSING

                                ;
                                ; PROCESS PARAMETERS
                                ;
000020                          $CTBAD::.BLKW   1               ; CTB ADDRESS (FOR RSX-11M+ ONLY)
000022                          $URM::  .BLKW   1               ; UNIBUS RUN MASK (FOR RSX-11M+ ONLY)
000024                          $RCDEV::.BLKW   1               ; RE-CONFIGURATION DEVICE (FOR RSX-11M+ ONLY)
000026                          $FEATR::.BLKW   1               ; CURRENT LLC DEVICE FEATURES WORD
000030                          $$EXT:: .BLKW   1               ; PROCESS EXTENSION IN 32. WORD BLOCKS
                                                                ; FOR DDM'S - NUMBER OF CONTROLLERS IN PROCESS SPACE
000032                          .NN::   .BLKW   1               ; NUMBER OF NODES MAXIMUM
```

```
                                        .SBTTL  $SCIR - SET CIRCUIT

                                ;+
                                ;
                                ; $SCIR - SET CIRCUIT
                                ;
                                ; INPUTS:
                                ;       $PDVA - PDV ADDRESS
                                ;       $SLTA - SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       CIRCUIT DATA BASE IS SET UP FROM CETAB
                                ;
                                ;-
000636  016701  000062'         $SCIR:  MOV     $PDVA,R1        ; GET DLM'S PDV ADDRESS
000642  004767  177626                  CALL    AIC             ; IS DLM ALREADY IN CORE?
000646  001016                          BNE     10$             ; IF NE, YES
000650  012767  003000  000014'         MOV     #MS.INC!MS.EXT,$MISS    ; LOOK FOR PROCESS INCREMENT
000656  004767  000000G                 CALL    $NLCFG          ; ..IN CETAB
000662  103427                          BCS     101$            ; IF CS, FAILED
000664  004767  000000G                 CALL    $XFER           ; TRANSFER DLM INTO CORE
000670  103424                          BCS     101$            ; IF CS, FAILURE
000672  052767  020000  001232'         BIS     #FL.PX2,$FLAGS  ; INDICATE SUCCESSFUL DDM TRANSFER
000700  010160  000000G                 MOV     R1,Z.PCB(R0)    ; INDICATE PROCESS IS LOADED
000704  016700  000136'         10$:    MOV     $SLTA,R0        ; GET SLT ADDRESS
000710  032760  040000  000000          BIT     #LF.RDY,L.FLG(R0) ; HAS LINE TABLE BEEN EXPANDED?
000716  001003                          BNE     20$             ; BR IF YES
000720  004767  000000G                 CALL    $TMP            ; EXPAND LINE TABLE
000724  103406                          BCS     101$
000726  052767  010000  000014' 20$:    BIS     #MS.CIR,$MISS   ; INDICATE CIRCUIT PARAMETERS ARE MISSING
000734  004767  000000G                 CALL    $NLCFG          ; LOOK FOR CIRCUIT IN CETAB
000740  000207                          RETURN

000742  000167  000000G         101$:   JMP     $XLIN           ; DO A PARTIAL CLEAR LINE
        000000'                 .END    $NTL
```

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 10

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES	
CCBDF$	#4-59	4-62
SLTDF$	#4-59	4-61

FILEID**NTLRQB

NTLRQB
CHKPP -

348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388

NTLRQB
SPRO -

```
                                        .SBTTL  CHKPP - CHECK PROCESS PARAMETERS

                                ;+
                                ; CHKPP - CHECK PROCESS PARAMETERS
                                ;
                                ; INPUTS:
                                ;       LS.OPT SET IF ANY OPTIONAL PARAMETERS
                                ;       R3=REQUEST BLOCK POINTER
                                ;               .BLKW   1       ; LO.LIN - NUMBER OF LINE TABLES IN PROCESS SPACE
                                ;               .BLKW   1       ; LO.CON - NUMBER OF CONTROLLERS IN PROCESS SPACE
                                ;               .BLKW   2       ; LO.PAR - PARTITION NAME
                                ;
                                ; OUTPUTS:
                                ;       R3=DESTROYED
                                ;       $$INC=LINE TABLE COUNT
                                ;       $$EXT=CONTROLLER TABLE COUNT IF NON-LLC
                                ;       $$PAR=PARTITION NAME
                                ;
                                ;-

000262  012767  000000C 000000G CHKPP:  MOV     #MS.INC!MS.EXT,$MISS    ; ASSUME PROCESS EXTENSION FIELDS ABSENT
000270  032767  000400  000004G         BIT     #LS.OPT,$$LRB+4 ; ANY OPTIONAL PARAMETERS INCLUDED?
000276  001002                         BNE     10$             ; IF NE, YES
000300  012703  000116'                MOV     #NULL,R3        ; IF NOT, USE NULL PARAMETERS
000304  012367  000000G         10$:   MOV     (R3)+,$$INC     ; SET LINE TABLE COUNT
000310  001403                         BEQ     20$             ; IF EQ, LINE TABLE COUNT NOT PRESENT
000312  042767  000000G 000000G         BIC     #MS.INC,$MISS   ; ELSE, NO LONGER MISSING
000320  005067  000000G         20$:   CLR     $$EXT           ; CLEAR LLC PROCESS EXTENSION
000324  032767  000000G 000000G         BIT     #FL.LLC,$FLAGS  ; IS THIS AN LLC ?
000332  001006                         BNE     30$             ; IF NE, YES
000334  011367  000000G                MOV     (R3),$$EXT      ; SET CONTROLLER TABLE COUNT
000340  001403                         BEQ     30$             ; IF EQ, CONTROLLER COUNT NOT PRESENT
000342  042767  000000G 000000G         BIC     #MS.EXT,$MISS   ; ELSE, NO LONGER MISSING
000350  005723                 30$:    TST     (R3)+           ; SKIP CONTROLLER TABLE COUNT
000352  012367  000000G                MOV     (R3)+,$$PAR     ; SET PARTITION NAME
000356  011367  000002G                MOV     (R3),$$PAR+2    ;
000362  005067  000000G                CLR     $$DEV           ; INDICATE NO DEVICE NAME (YET)
000366  005067  000000G                CLR     $$TBL           ; AND NO DISPATCH TABLE ADDRESS
000372  005067  000000G                CLR     $$LINX          ; CLEAR LOGICAL LINK COUNT
000376  005067  000000G                CLR     $$NODC          ; CLEAR NODE COUNTERS COUNT
000402                                 RETURN
```

```
                        .SBTTL  CCEX - CHECK CLEAR CEX REQUEST BLOCK

                ;+
                ; CCEX - CHECK "CLEAR CEX" REQUEST BLOCK
                ;
                ; INPUTS:
                ;       $PDVA=PDV ADDRESS
                ;       $PDVNUM=NUMBER OF PDV ENTRIES
                ;       $SLTMA=SLT TABLE ADDRESS
                ;       $SLTNM=NUMBER OF SLT ENTRIES
                ;
                ; OUTPUTS:
                ;       C-BIT=SUCCESS/FAILURE
                ;
                ;
                ;-

002156          CCEX:
002156  004567  000000G         JSR     R5,$SAVRG       ; SAVE R3,R4,R5

                        .IF DF  I$$AS

                        CLR     R0
                        MOV     #'NS,R4         ; GET NS DEVICE NAME IN ASCII
                        CLR     R5              ; UNIT NUMBER 0
                        CALL    ..FDEV          ; LOOK FOR PUD ENTRY
                        BCS     5$              ; BR IF NOT FOUND
                        BITB    #UM.RT,U.MN(R5) ; IS NS: STILL MOUNTED?
                        BEQ     5$              ; BR IF NO
                        MOV     U.DN(R5),R1     ; GET NAME OF MOUNTED DEVICE
                        EMSG$R  Z4              ; INDICATE DEVICE STILL MOUNTED
                5$:
                        .ENDC

002162  005000          CLR     R0              ; USED AS AN ERROR INDICATOR
002164  005067  000126' CLR     PDVX            ; START WITH PDV INDEX = 0
002170  017705  000000G MOV     @PDVTA,R5       ; GET PDV TABLE ADDRESS

002174  017767  000000G 000134' MOV     @PDVNM,PDVNUM   ; GET NUMBER OF PDV ENTRIES
002202  001002          BNE     10$             ; IF NON-ZERO, ENTRIES
002204  000167  000444  JMP     190$            ; IF ZERO, NO ENTRIES
002210  012504          10$:    MOV     (R5)+,R4        ; GET ADDRESS OF NEXT PDV
002212  001566          BEQ     145$            ; ALLOW FOR BLANK ENTRIES

002214  005764  000000G TST     Z.PCB(R4)       ; IS IT LOADED?
002220  001563          BEQ     145$            ; IF EQ, NO
002222  010467  000000G MOV     R4,$PDVA        ; SAVE PDV ADDRESS FOR ERROR MESSAGE
002226  132764  000000G 000000G BITB    #FL.LLC,Z.FLG(R4) ; IS THIS AN LLC?
002234  001541          BEQ     130$            ; IF EQ, NO
002236  005002          CLR     R2              ; GET THE CHANNEL COUNT
002240  156402  000000G BISB    Z.LLN(R4),R2    ;
002244  001447          BEQ     50$             ; IF EQ, NONE
002246  006302          ASL     R2              ; CONVERT TO A WORD INDEX
002250  010401          MOV     R4,R1           ; GET PDV ADDRESS
002252  062701  000000G ADD     #Z.MAP,R1       ; POINT AT MAPPING TABLE
002256  060102          ADD     R1,R2           ; GIVING END OF MAPPING TABLE
```

Symbol table

```
A$$CHK= 000000          GS$WRD= 000000          LR.UNT  000005          M$$NET= 000000          US.MNT= ****** GX
A$$CPS= 000000          I$$PAR= 000000          LS.AST= 100000          M$$OVR= 000000          U.CW1 = ****** GX
A$$PRI= 000000          I$$RDN= 000000          LS.CEX= 000001          NTLPT = ****** GX       U.STS = ****** GX
A$$TRP= 000000          KMCPF   002120R         LS.CIR= 000010          NULL    000116R     002 V$$CTR= 001000
CCEX    002156R         K$$CNT= 177546          LS.CXQ= 010000          N$$ACC= 000001          X$$DBT= 000000
CEXLD   000164R         K$$CSR= 177546          LS.CX5= 000400          N$$BUF= 000001          ZF.DDM= ****** GX
CHKLP   000730R         K$$LDC= 000000          LS.DLM= 004000          N$$LDV= 000001          ZF.DLC= ****** GX
CHKPP   000262R         K$$TPS= 000074          LS.ECH= 001000          N$$MCP= 000001          ZF.KMX= ****** GX
CLIN    002054R         LBITS   000136R     002 LS.FDX= 004000          N$$MLL= 000001          ZF.LMC= ****** GX
CLRPP   001244R         LD$LP = 000000          LS.HDX= 010000          N$$MOV= 000010          ZF.MUX= ****** GX
CPRO    001306R         LF.ACT= 100000          LS.LIN= 000003          N$$NCT= 000001          Z.DAT = ****** GX
C$$CKP= 000000          LF.BRO= 000400          LS.LMC= 000007          N$$PEM= 000001          Z.FLG = ****** GX
C$$ORE= 000400          LF.BWT= 000007          LS.NTI= 004000          PBITS   000146R     002 Z.LLN = ****** GX
C$$RSH= 177564          LF.ENA= 002000          LS.ON = 000011          PDVA    000204R         Z.MAP = ****** GX
DEVHD = ****** GX       LF.LPB= 001000          LS.OPT= 000400          PDVNM = ****** GX       Z.NAM = ****** GX
DEV7    003532R         LF.MDC= 000100          LS.PRO= 000002          PDVNUM  000134R     002 Z.PCB = ****** GX
DV.MNT= ****** GX       LF.MFL= 004000          LS.PWF= 040000          PDVTA = ****** GX       $BIAS = ****** GX
D$$BUG= 177514          LF.MTP= 000020          LS.TOP= 002000          PDVX    000126R     002 $BITS   003016R
D$$ISK= 000000          LF.PAC= 000200          LS.UNF= 020000          P$$P45= 000000          $CBOMG= ****** GX
D$$L11= 000001          LF.RDY= 040000          LS.X25= 000012          P$$WRD= 000000          $C5TA = ****** GX
D$$YNC= 000000          LF.REA= 010000          LS.11D= 004000          P.REL = ****** GX       $DDMPV= ****** GX
D$$YNM= 000000          LF.SER= 000040          LX.CEX= 000004          Q$$OPT= 000010          $DLCPV= ****** GX
D.DSP = ****** GX       LF.TIM= 000010          LX.LIN= 000006          REP7    003420R         $ERRZP  001406R     002
D.NAM = ****** GX       LF.UNL= 020000          LX.PRO= 000005          REP7A   003422R         $ERRZR  001264R     002
D.PCB = ****** GX       LF.X2P= 000000          L$$ASG= 000000          REP7B   003564R         $ERRZ1  001120R     002
D.UCB = ****** GX       LN.CLO= 000000          L$$DRV= 000000          R$$DER= 000000          $ERRZ3  001150R     002
D.UCBL= ****** GX       LN.DUM= 000005          L$$P11= 000001          R$$K11= 000001          $ERRZ4  001162R     002
D.UNIT= ****** GX       LN.LOA= 000004          L$$11R= 000000          R$$SND= 000000          $ERRZ5  001330R     002
ERRBUF  000170R     002 LN.LOO= 000003          L.COST  000015          R$$11M= 000000          $ERRZ6  001354R     002
ERRMSG  000160R     002 LN.OAU= 000003          L.CTL   000012          SCEX    000070R         $ERRZ7  001224R     002
E$$XPR= 000000          LN.OFF= 000001          L.CVA   177776          SCIR    002656R         $ERR86  001062R     002
FE.CEX= ****** GX       LN.ON = 000000          L.DDM   000002          SF.ACT= 000200          $ERR87  001014R     002
FL.CEX= ****** GX       LN.OOP= 000004          L.DDS   000004          SF.ENA= 000100          $ERR88  000752R     002
FL.DDM= ****** GX       LN.OPE= 000001          L.DLC   000003          SF.LPB= 000004          $ERR90  000716R     002
FL.DLC= ****** GX       LN.REF= 000002          L.DLM   000006          SF.MFL= 000040          $ERR91  000660R     002
FL.ECH= ****** GX       LN.SER= 000002          L.DLS   000010          SF.PAC= 000020          $ERR92  000616R     002
FL.FDX= ****** GX       LN.STA= 000017          L.FLG   000000          SF.REA= 000010          $ERR93  000552R     002
FL.HDX= ****** GX       LN.SUB= 000360          L.KRBA  000016          SF.SER= 000001          $ERR94  000516R     002
FL.KMX= ****** GX       LN.TRI= 000006          L.LEN = 000022          SF.SVC= 000002          $ERR95  000462R     002
FL.LLC= ****** GX       LO.CON  000006          L.MPF   000022          SF.UNL= 000040          $ERR96  000424R     002
FL.LMC= ****** GX       LO.CSR  000006          L.NMST  000020          SLIN    001256R         $ERR97  000366R     002
FL.MUX= ****** GX       LO.CTL  000014          L.NSTA  000014          SLTA    000464R         $ERR98  000330R     002
FL.NTI= ****** GX       LO.LIN  000004          L.OWNR  000021          SLTMA = ****** GX       $ERR99  000276R     002
FL.TOP= ****** GX       LO.PAR  000010          L.UNT   000013          SLTNM = ****** GX       $FLAGS= ****** GX
FMASK = ****** GX       LO.PRI  000010          MS.CSR= ****** GX       SPRO    000404R         $FPDV   003050RG
FMAX  = 000010          LO.SCR  000014          MS.EXT= ****** GX       SPSAV   000132R     002 $FSLT   003140RG
FMT7    001444R     002 LO.STA  000015          MS.INC= ****** GX       S$$WRG= 000000          $FUNC = ****** GX
FMT7A   001445R     002 LO.VCT  000012          MS.LIN= ****** GX       S$$YSZ= 007600          $MAPX = ****** GX
FMT7B   001511R     002 LR.AST  000026          MS.LLC= ****** GX       S.COST  000001          $MISS = ****** GX
FM.7  = 000000          LR.CIR  000002          MS.MTP= ****** GX       S.FLG   000000          $MXVCT= ****** GX
FM.7A = 000000          LR.CTL  000004          MS.PRI= ****** GX       S.LEN   000004          $NAME = ****** GX
FM.7B = 000000          LR.EFN  000004          MS.SEC= ****** GX       S.NMSI  000002          $NLCSR= ****** GX
FTAB    000000R     002 LR.LIN  000002          MS.VCT= ****** GX       S.OWNR  000003          $NLRQB  000000RG
F$$LVL= 000001          LR.PRM  000030          M$$CRB= 000124          T$$KMG= 000000          $PCB  = ****** GX
G$$TPP= 000000          LR.PRO  000002          M$$CRX= 000000          T$$MIN= 000000          $PDVA = ****** GX
G$$TSS= 000000          LR.STS  000000          M$$FCS= 000000          UNITS   000130R     002 $RERR   003332RG
G$$TTK= 000000          LR.TCB  000002          M$$MGE= 000000          US.BSY= ****** GX       $SAVAL= ****** GX
```


Symbol table

```
                                ;+
                                ; $DFN - SETUP DEFAULT FILE NAME
                                ;
                                ; INPUTS:
                                ;       R0=RAD50 FILE NAME, FIRST HALF
                                ;       R1=RAD50 FILE NAME, SECOND HALF ($DFN2 ONLY)
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;-
000000                          $DFN::
000000  005C01                          CLR     R1              ; NO SECOND HALF
000002                          $DFN2::
000002  010146                          MOV     R1,-(SP)        ; SAVE SECOND HALF
000004  010001                          MOV     R0,R1           ; COPY FIRST HALF
000006  012700  000112'                 MOV     #$DFNAM,R0      ; GET DEFAULT FILE NAME ADDRESS
000012                                  CALL    10$             ; CONVERT TO ASCII
000016  012601                          MOV     (SP)+,R1        ; RESTORE SECOND HALF
000020  001407                          BEQ     20$             ; IF ZERO, USE FIRST HALF ONLY
000022                          10$:    CALL    $C5TA           ; CONVERT TO ASCII
000026  122740  000040          15$:    CMPB    #' ,-(R0)       ; SKIP TRAILING BLANKS
000032  001775                          BEQ     15$             ;       ...
000034  005200                          INC     R0              ;       ...
000036  105010                          CLRB    (R0)            ; MAKE IT AN ASCIZ STRING
000040                          20$:    RETURN
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  PDVDF$



                                ;***
                                ; GLOBAL LIBRARY SYMBOLS
                                ;***

 000000                                 PDVDF$  <=>,<:>,LIST    ; PROCESS DESCRIPTION VECTOR OFFSETS
                                ;
                                ;       PROCESS DESCRIPTOR VECTOR OFFSETS AND BIT DEFINITIONS
                                ;

 000000                         Z.DSP:: .BLKW   1       ; RELOCATION BIAS OF DISPATCH TABLE
 000002                                 .BLKW   1       ; VIRTUAL ADDRESS OF DISPATCH TABLE

                                        .IF DF  X$$MCB
                                Z.QUE:: .BLKW   1       ; ADDRESS OF SCHEDULER LEVEL QUEUE
                                        .ENDC
 000004                         Z.NAM:: .BLKW   1       ; PROCESS NAME (RAD50)
 000006                         Z.LLN:: .BLKB   1       ; # OF LOGICAL LINES (LLC'S ONLY)
 000007                         Z.SCH:: .BLKB   1       ; PROCESS PRIORITY (COMPLEMENT)
 000010                         Z.FLG:: .BLKW   1       ; FLAGS WORD
                                        .IF DF  X$$MCB!R$$11M!R$$11S
 000012                         Z.PCB:: .BLKW   1       ; PCB OF LOADED PROCESS PARTITION
                                        .ENDC
                                        .IF DF  R$$11D!I$$AS
                                Z.GCD:: .BLKW   1       ; ADDRESS OF GCD NODE( 0 = PROCESS NOT RESIDENT )
                                        .ENDC
 000014                         Z.AVL:: .BLKW   1       ; POINTER TO FIRST FREE BLOCK IN PROCESS SPACE
 000016                         Z.LEN== .-Z.DSP         ; LENGTH OF PDV
 000016                         Z.DAT:: .BLKW   1       ; BIAS OF LLC DATA BASE
 000020                         Z.MAP::                 ; DISPLACEMENT OF LLC MAPPING TABLE
                                ;
                                ; FLAGS WORD BIT DEFINTIONS
                                ;
                 000000         ZF.X3P==    0           ; PROCESS REQUIRES PARAMETERS FROM X3P$DF MACRO
                 000001         ZF.DDM==    1           ; PROCESS IS A DDM
                 000002         ZF.DLC==    2           ; PROCESS IS A DLC
                 000004         ZF.LLC==    4           ; PROCESS IS A LLC
                 000010         ZF.MFL==   10           ; PROCESS IS MARKED FOR LOAD ( WITH NS )
                 000020         ZF.KMX==   20           ; PROCESS IS A DDM FOR A KMC DEVICE
                 000040         ZF.MUX==   40           ; PROCESS IS A DDM FOR A MULTIPLEXER DEVICE
                 000100         ZF.LMC==  100           ; PROCESS REQUIRES MICRO-CODE TO BE LOADED
                 000200         ZF.TIM==  200           ; PROCESS REQUESTS TIMER SUPPORT
                 000400         ZF.MTM==  400           ; PROCESS REQUIRES TIMER ENTRY ON EVERY ACTIVE PROCESSOR
                                                        ; (RSX-11M+ MULTI-PROCESSOR SYSTEMS ONLY)
                 001000         ZF.COU==1000            ; LLC/DLC PROCESS SUPPORTS COUNTERS
                 002000         ZF.PSE==2000            ; PROCESS IS A PSEUDO DDM/DLC
                 004000         ZF.DIA==4000            ; PROCESS REQUIRES DIAGNOSTICS MICRO-CODE LOAD
                 010000         ZF.SLI==10000           ; PROCESS SUPPORTS SYSTEM LEVEL INTERFACE
                 020000         ZF.MAN==20000           ; PROCESS SUPPORTS NETWORK MANAGEMENT REQUESTS
                 040000         ZF.INI==40000           ; PROCESS SUPPORTS INITIALIZATION AND TERMINATE
                 100000         ZF.DVP==100000  ; PROCESS ALWAYS RUNS AT IT'S PRIORITY
                                ;
```

```
                                        ;
                                        ; ERROR MESSAGE BUFFER
                                        ;
000354  015                     ERRMSG: .BYTE   15
000355  116     124     114             .ASCII  'NTL '
000361                          ERRBUF: .BLKB   73.

                                        ;
                                        ; ERROR MESSAGE PREFIXES
                                        ;
000472  055     055     040     WPFX:   .ASCII  '-- *WARNING* '
000507  055     055     040     EPFX:   .ASCIZ  '-- '

                                        .EVEN

                                        ;
                                        ; ERROR MESSAGES
                                        ;
                                        .ENABL  LC

000514                                  NTLER$  ,13,2,$TERR,$REP.C,$ELINE,<Binary Buffer Overflow>
000554                                  NTLER$  ,21,2,$TERR,$REP.C,$ELINE,<DDM Operator>
000602                                  NTLER$  ,79,2,$TERR,$REP.C,$ELINE,<DLC Operator>
000630                                  NTLER$  ,82,2,$TERR,$REP.C,$ELINE,<Operator May Not Occur Twice>
000676                                  NTLER$  ,83,2,$TERR,$REP.C,$ELINE,<Primary Template Operator>

                                        .DSABL  LC

                                        ;
                                        ; ERROR MESSAGE FORMAT STRING
                                        ;
                                        .ENABL  LC
                                        .NLIST  BEX
000740  052     040     124     FMT2:   .ASCIZ  '* Template -- '
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

000000                                  .PSECT
```

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 2

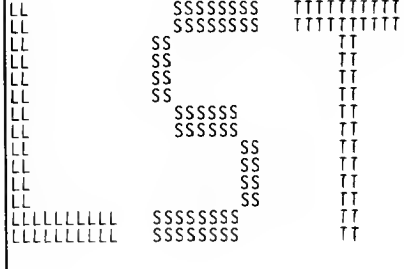
SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
$CLBUF	= ****** GX	*7-278							
$CLDEQ	= ****** GX	6-260							
$CLIOS	= ****** GX	7-279							
$CLSIZ	= ****** GX	*7-278							
$CVAL	000136 RG	#5-154							
$DIFF	000112 RG	#5-147							
$EFNLL	000326 RG	#9-326							
$ELINE	000332 RG	5-189	5-190	5-191	5-192	5-193	6-251	#9-328	
$ERR13	000514 R	#5-189	10-371						
$ERR21	000554 R	#5-190	12-415						
$ERR79	000602 R	#5-191	13-430						
$ERR82	000630 R	#5-192	14-459						
$ERR83	000676 R	#5-193	14-460						
$EVEN	000470 RG	#11-393							
$FILEN	000100 RG	#5-146							
$FILEX	= ****** GX	8-307							
$FLAGS	= ****** GX	6-249	12-413	13-428	15-480	15-484			
$FLAG1	000000 RG	#5-69	*6-235	10-367	*10-372	10-375	14-448	*14-450	*15-487
$FLAG2	000002 RG	#5-91	6-242	*6-245	6-258	*6-261			
$FNOUT	= ****** GX	9-327							
$FNR50	000130 RG	#5-153							
$GCLEP	= ****** GX	7-278							
$ICBAD	000072 RG	#5-136							
$ICBSZ	000074 RG	#5-137							
$JSROF	= 000012 G	#5-138							
$LBADR	000140 RG	#5-155							
$LBIAS	000144 RG	#5-158							
$LDA	000054 RG	#5-125							
$LEFT	000010 RG	#5-105	*10-369						
$LFLD	000060 RG	#5-127							
$MPLD	000056 RG	#5-126							
$MXADD	000020 RG	#5-113							
$MXLEN	000026 RG	#5-116							
$MXSIZ	000022 RG	#5-114							
$NALL	000030 RG	#5-117							
$NLIB	000031 RG	#5-118							
$NVECT	000070 RG	#5-135							
$OFF	000024 RG	#5-115							
$PRAPR	= 120000 G	#5-139							
$PTR	000006 RG	#5-104	10-377	*10-378					
$REP.C	= ****** GX	5-189	5-190	5-191	5-192	5-193			
$R01	000404 RG	#10-366							
$R02	000376 RG	#10-364	11-396						
$R03	000370 RG	#10-362							
$R04	000362 RG	#10-360							
$R05	000354 RG	#10-358							
$R07	000346 RG	#10-356							
$SAVAL	= ****** GX	6-241	15-489						
$SIZE	000014 RG	#5-111	6-244	11-394	*11-400				
$SLTA	= ****** GX	6-225							
$SLTSV	000012 RG	#5-110	*6-225						
$SZPTR	000032 RG	#5-119	*6-243	6-244	*6-262	14-452	14-452		

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 3

FILEID**NTLGET

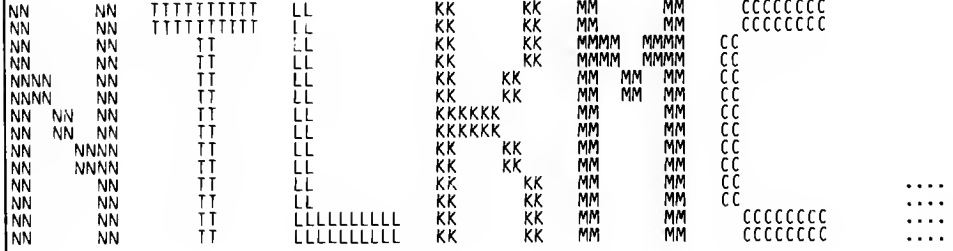
NN NN TTTTTTTTTT LL GGGGGGGG EEEEEEEEEE TTTTTTTTTT
NN NN TTTTTTTTTT LL GGGGGGGG EEEEEEEEEE TTTTTTTTTT
NN NN TT LL GG EE TT
NN NN TT LL GG EE TT
NNNN NN TT LL GG EE TT
NNNN NN TT LL GG EE TT
NN NN NN TT LL GG EEEEEEEE TT
NN NN NN TT LL GG EEEEEEEE TT
NN NNNN TT LL GG GGGGGG EE TT
NN NNNN TT LL GG GGGGGG EE TT
NN NN TT LL GG GG EE TT ....
NN NN TT LL GG GG EE TT ....
NN NN TT LLLLLLLLLL GGGGGG EEEEEEEEEE TT ....
NN NN TT LLLLLLLLLL GGGGGG EEEEEEEEEE TT ....

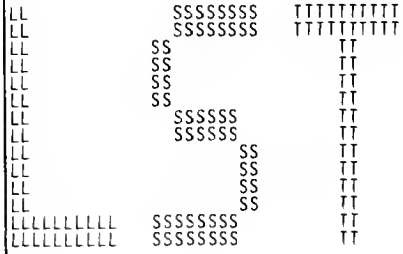


NTLGET - GET SEND/RECEIVE DATA MACRO V05.03b Saturday 29-Jun-85 00:31

Table of contents

FILEID**NTLKMC


NN NN TTTTTTTTTT LL KK KK MM MM CCCCCCCC
NN NN TTTTTTTTTT LL KK KK MM MM CCCCCCCC
NN NN TT LL KK KK MMMM MMMM CC
NN NN TT LL KK KK MMMM MMMM CC
NNNN NN TT LL KK KK MM MM MM CC
NNNN NN TT LL KK KK MM MM MM CC
NN NN NN TT LL KKKKK MM MM CC
NN NN NN TT LL KKKKK MM MM CC
NN NNNN TT LL KK KK MM MM CC
NN NNNN TT LL KK KK MM MM CC
NN NN TT LL KK KK MM MM CC ....
NN NN TT LL KK KK MM MM CCCCCCCC ....
NN NN TT LLLLLLLLLL KK KK MM MM CCCCCCCC ....
NN NN TT LLLLLLLLLL KK KK MM MM CCCCCCCC ....



NTLKMC - LOAD KMC MICRO-CODE MACRO V05.03b Saturday 29-Jun-85 00:31

Table of contents

445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501

```
                        MOV     #7,-(SP)                ;B-7 ENTRIES PER LIST
                ;
                ; FIND A BLANK ENTRY AND FILL IT IN
                ;
15$:            TST     (R0)                    ;B-IS THIS ENTRY EMPTY?
                BNE     20$                     ;B-BR IF NO
                MOV     KMCNAM,(R0)+            ;B-FILL IN RAD50 NAME
                MOV     KMCCSR,(R0)+            ;B-FILL IN CSR ADDRESS
                MOV     KMCNUM,(R0)+            ;B-SET CTL NUM AND MICRO-CODE LEN
                CLR     (R0)                    ;B-NO LOCAL MICRO-CODE COPY
                TST     (SP)+                   ;B-DUMP ENTRY COUNT
                BR      50$                     ;B-DONE
20$:            ADD     #8.,R0                  ;B-SKIP REST OF ENTRY
                DEC     (SP)                    ;B-COUNT ENTRIES
                BGT     15$                     ;B-BR IF MORE IN THIS BLOCK
                TST     (SP)+                   ;B-DUMP ENTRY COUNT
                BR      10$                     ;B-CHECK NEXT BLOCK
                ;
                ; RAN OUT OF BLOCKS TO CHECK, A NEW BLOCK MUST BE ALLOCATED
                ;
45$:            .IF DF  M$$MGE
000376 011167 000012'   MOV     (R1),KMCMP1             ;M-SAVE MAPPING OF LINK TO NEW BLOCK
                .ENDC
000402 010367 000014'   MOV     R3,KMCPN1               ;B-SAVE POINTER TO LINK WORD
                ;
                ; MAPPED SYSTEMS ALWAYS COME HERE AFTER A LIST SEARCH
                ;
50$:            .IF DF  M$$MGE
                 .IF DF  R$$11M
000406          RETURN                          ;M-RETURN TO 55$ FROM SWSTK$
                .IFF
                CALL    @(SP)+                  ;M-RESTORE MAPPING VIA MAPX
                 .ENDC
                .ENDC
000410 005767 000014'   55$:    TST     KMCPN1                  ;B-WAS INFO ENTERED INTO LIST?
000414 001451           BEQ     70$                     ;B-BR IF YES, FINISHED
                ;
                ; A BLOCK MUST BE ALLOCATED FOR A NEW LIST SEGMENT (RAN OUT OF ROOM)
                ;
000416 012701 000001            MOV     #KM.ALL,R1              ;B-AMOUNT TO ALLOCATE
000422                          CALL    $ALN18                  ;B-GET    K FROM NON-UMR SPACE
000426 103017                   BCC     60$                     ;B-BR    NO ALLOCATION FAILURE
000430 016701 000000G           MOV     $NAME,R1                ;B-GET THE DEVICE/PDV NAME
000434 012700 000006'           MOV     #KMCDEV,R0              ;B-PLACE ASCII VERSION HERE
000440                          CALL    $C5TA                   ;B-CONVERT RAD50 TO ASCII
000444 105010                   CLRB    (R0)                    ;B-MAKE ASCIZ
000446 016767 000004' 0000 5'   MOV     KMCNUM,KMCER1           ;B-USE CONTROLLER NUM FOR THIRD ARG
000454                          EMSG$   3J                      ;B-ALLOCATION FAILURE ERROR
000462 000261                   SEC                             ;B-INDICATE FAILURE
000464 000425                   BR      70$
                ;
                ; GOT A NEW BLOCK - INITIALIZE IT AND LINK IT IN
                ;
60$:            .IF DF  M$$MGE
                 .IF DF  R$$11M
000466          SWSTK$  70$                     ;* ENTER KERNEL MODE
```

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
FMT10A	000512 R	8-176	8-177	8-178	8-179	8-180	8-181	#8-194		
FMT10B	000540 R	8-182	#8-195							
FMT10C	000615 R	8-183	#8-196							
FM.10A	= 000000	#8-176	#8-177	#8-178	#8-179	#8-180	#8-181			
FM.10B	= 000000	#8-182								
FM.10C	= 000000	#8-183								
KMCBUF	000000 R	#8-210	15-609							
KMCCSR	000002 R	#7-132	*11-311	11-348	*12-416	12-452	12-511	*17-703	20-854	
KMCDEV	000006 R	#7-134	*10-298	*10-304	12-489	13-557	20-849			
KMCEBF	000024 R	#8-155	18-763							
KMCEFT	001234 R	8-176	8-177	8-178	8-179	8-180	8-181	8-182	8-183	#18-748
KMCEMS	000034 R	#8-156	18-761							
KMCER1	000016 R	#7-139	*12-492	*17-704	20-857					
KMCER2	000020 R	#7-140	*17-705	21-886						
KMCER3	000022 R	#7-141	*17-706	21-889						
KMCEX	001232 R	16-676	#17-715							
KMCFEC	001312 R	8-176	#19-796							
KMCFEO	001320 R	8-177	8-178	8-179	8-180	8-181	#19-799			
KMCLD	000542 R	9-245	11-349	#13-542						
KMCLL	000234 R	9-247	#12-410							
KMCMP1	000012 R	#7-136	*10-273	10-279	*10-288	*10-293	11-321	*12-466	12-519	
KMCNAM	000000 R	#7-131	*10-300	11-323	*12-412	12-451	12-510			
KMCNUM	000004 R	#7-133	*11-312	*12-414	*12-415	12-453	12-492	12-512		
KMCPN1	000014 R	#7-138	10-282	*11-313	*12-411	*12-468	12-480	12-522		
KMCRL1	001372 R	8-182	8-183	#20-846						
KMCRL2	001444 R	8-182	#21-882							
KM.ALL	= 000001	#7-144	12-465							
LF.ACT	= 100000	#5-60								
LF.BRO	= 000400	#5-60								
LF.BWT	= 000007	#5-60								
LF.ENA	= 002000	#5-60								
LF.LPB	= 001000	#5-60								
LF.MDC	= 000100	#5-60								
LF.MFL	= 004000	#5-60								
LF.MTP	= 000020	#5-60								
LF.PAC	= 000200	#5-60								
LF.RDY	= 040000	#5-60								
LF.REA	= 010000	#5-60								
LF.SER	= 000040	#5-60								
LF.TIM	= 000010	#5-60								
LF.UNL	= 020000	#5-60								
LF.X2P	= 000000	#5-60								
LN.CLO	= 000000	#5-60								
LN.DUM	= 000005	#5-60								
LN.LOA	= 000004	#5-60								
LN.LOO	= 000003	#5-60								
LN.OAU	= 000003	#5-60								
LN.OFF	= 000001	#5-60								
LN.ON	= 000000	#5-60								
LN.OOP	= 000004	#5-60								
LN.OPE	= 000001	#5-60								
LN.REF	= 000002	#5-60								

NTLPRO
AXSYM

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 2

CREF 04.00

NTLPRO
$CTSYM

```
                                        .SBTTL  AXSYM - AUX SPECIAL SYMBOLS
                                ;+
                                ; *** - AXSYM - AUX SPECIAL SYMBOLS
                                ;
                                ; INPUTS:
                                ;       $NAME = RAD50 PROCESS NAME
                                ;       $SQSRV = ADDRESS OF $SQSRV ROUTINE IN AUX PROCESS
                                ;       PR100M = ADDRESS OF 100 MSEC TIMER SERVICE ROUTINE IN AUX PROCESS
                                ;       PR1SC = ADDRESS OF 1 SEC TIMER SERVICE ROUTINE IN AUX PROCESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       FORK BLOCK AND TIMER BLOCKS INITIALIZED
                                ;
                                ;-
000234  022767  004640  000000G AXSYM:  CMP     #^RAUX,$NAME            ; IS THIS PROCESS AUX ?
000242  000241                         CLC                             ; ASSUME NOT
000244  001103                         BNE     20$                     ; IF NE, NO .. DO NOTHING

000246  016767  000000G 000000'        MOV     CMFRK,TEMP              ; STORE $CMFRK ADDESSS
000254  062767  000010  000000'        ADD     #10,TEMP                ; GET $CMFRK+10 ADDERSS
000262  016777  000000G 000000'        MOV     $BIAS,@TEMP             ; STORE BIAS FOR AUX PROCESS
000270  016767  000000G 000000'        MOV     T100C,TEMP              ; STORE $T100C ADDRESS
000276  062767  000014  000000'        ADD     #14,TEMP                ; GET $T100C+14 ADDRESS
000304  016777  000000G 000000'        MOV     $BIAS,@TEMP             ; STORE BIAS
000312  016767  000000G 000000'        MOV     T1SCL,TEMP              ; GET $T1SCL ADDRESS
000320  062767  000014  000000'        ADD     #14,TEMP                ; GET $T1SCL+14 ADDRESS
000326  016777  000000G 000000'        MOV     $BIAS,@TEMP             ; STORE BIAS

000334  005767  000072'                TST     SQBLK+2                 ; WAS $SQSRV DEFINED ?
000340  001444                         BEQ     10$                     ; IF EQ, NO .. ERROR
000342  016767  000000G 000000'        MOV     CMFRK,TEMP              ; GET $CMFRK ADDRESS
000350  016777  000100' 000000'        MOV     $SQSRV,@TEMP            ; INITIALIZE FORK BLOCK
000356  062767  000002  000000'        ADD     #2,TEMP                 ; GET $CMFRK+2 ADDRESS
000364  016777  000100' 000000'        MOV     $SQSRV,@TEMP            ;
000372  005767  000104'                TST     MILBLK+2                ; WAS PR100M DEFINED ?
000376  001425                         BEQ     10$                     ; IF EQ, NO .. ERROR

000400  016767  000000G 000000'        MOV     T100C,TEMP              ; GET $T100C ADDRESS
000406  062767  000012  000000'        ADD     #12,TEMP                ; GET $T100C+12 ADDRESS
000414  016777  000112' 000000'        MOV     PR100M,@TEMP            ; INITIALIZE 100 MSEC CLOCK BLOCK
000422  016767  000000G 000000'        MOV     T1SCL,TEMP              ; GET $T1SCL ADDRESS
000430  062767  000012  000000'        ADD     #12,TEMP                ; GET $T1SCL+12 ADDRESS
000436  016777  000124' 000000'        MOV     PR1SC,@TEMP             ; INITIALIZE 1 SEC CLOCK BLOCK
000444  005767  000116'                TST     SECBLK+2                ; WAS PR1SC DEFINED ?
000450  001001                         BNE     20$                     ; IF NE, YES .. OKAY
000452  000261                 10$:    SEC                             ; ELSE, INDICATE FAILURE
000454                         20$:    RETURN
```

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL VALUE REFERENCES

```
LN.ON    = 000000        #5-65
LN.OOP   = 000004        #5-65
LN.OPE   = 000001        #5-65
LN.REF   = 000002        #5-65
LN.SER   = 000002        #5-65
LN.STA   = 000017        #5-65
LN.SUB   = 000360        #5-65
LN.TRI   = 000006        #5-65
L.COST     000015        #5-65
L.CTL      000012        #5-65
L.CVA      177776        #5-65
L.DDM      000002        #5-65
L.DDS      000004        #5-65
L.DLC      000003        #5-65
L.DLM      000006        #5-65
L.DLS      000010        #5-65
L.FLG      000000        #5-65
L.KRBA     000016        #5-65
L.LEN    = 000022        #5-65
L.MPF      000022        #5-65
L.NMST     000020        #5-65
L.NSTA     000014        #5-65
L.OWNR     000021        #5-65
L.UNT      000013        #5-65
MILBLK     000102 R       6-200     #6-204     8-360
M$$MGE   = 000000         12-653     13-707     13-713     13-723
NODCTR     000004 R      #6-108     *13-700    13-709
NTLPT    = ******  GX     10-547
N$ENC      000042        *13-699    *13-708    *13-724
N$MHC1     000036        *12-649    *12-651    *12-654
N$MHC2     000044        *12-659    *12-660    *12-661
PBLK       000020 R      #6-155      10-530     10-543
PRSYM      000462 R       7-277     #10-464
PR1SC      000124 R      #6-210      8-368
PR100M     000112 R      #6-206      8-365
PSYM       000024 R      #6-158     *10-481    *10-484
PVAL       000030 R      #6-159      7-291
P.BLKS   = ******  GX      7-293      7-300
P.REL    = ******  GX      7-289      7-290
R$$EIS   = ******          7-297      7-301      12-641     13-693
R$$MPL   = ******          5-69       6-122      6-152      6-164      9-389      9-447      10-486     10-534     10-553
R$$11D   = ******          7-297      7-301      12-641     13-693
R$$11M   = 000000         12-653     13-707     13-713     13-723
SECBLK     000114 R       6-204     #6-208     8-369
SF.ACT   = 000200        #5-65
SF.ENA   = 000100        #5-65
SF.LPB   = 000004        #5-65
SF.MFL   = 000040        #5-65
SF.PAC   = 000020        #5-65
SF.REA   = 000010        #5-65
SF.SER   = 000001        #5-65
SF.SVC   = 000002        #5-65
```


NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 3

```
                        .MAXC:: .BLKW   1               ; MAXIMUM COST ALLOWED
                        .MAXH:: .BLKW   1               ; MAXIMUM HOPS ALLOWED
                        .ROUTM::.BLKW   1               ; ROUTING TIMER INTERAL
                        .NA::   .BLKW   1               ; NUMBER OF AREAS IN NETWORK
                        .NBRA:: .BLKW   1               ; NUMBER OF BROADCAST ROUTER ADJACENCIES
                        .NBEA:: .BLKW   1               ; NUMBER OF BROADCAST ENDNODE ADJACENCIES
                        .AMAXH::.BLKW   1               ; AREA MAXIMUM HOPS
                        .AMAXC::.BLKW   1               ; AREA MAXIMUM COST
                        .ROUTB::.BLKW   1               ; BROADCAST ROUTING TIMER
                        .RPRI:: .BLKW   1               ; ROUTER PRIORITY
                        ..SCSR::.BLKW   1               ; SECONDARY CSR

                        ;
                        ; ********* NOTE: THE FOLLOWING DATA STRUCTURES ARE ************
                        ;                 DUPLICATED IN TMPEX6, SO ANY CHANGE MADE
                        ;                 HERE MUST BE REFLECTED IN TMPEX6 ALSO !!
                        ;
                        $PDVA:: .BLKW   1               ; PDV ADDRESS
                        $NAME:: .BLKW   1               ; RAD50 PDV NAME
                        $BIAS:: .BLKW   1               ; RELOCATION BIAS
                        $PCB::  .BLKW   1               ; PCB ADDRESS
                        $$INC:: .BLKW   1               ; NUMBER OF LINE TABLES TO PUT IN PROCESS SPACE
                        $$PAR:: .BLKW   2               ; RAD50 PARTITION NAME
                        $$DEV:: .BLKW   1               ; ASCII DEVICE NAME
                        $$TBL:: .BLKW   1               ; DEVICE DISPATCH TABLE ADDRESS
                        $$LINX::.BLKW   1               ; LOGICAL LINK COUNT
                        $CTIM:: .BLKW   1               ; COUNTER TIMER VALUE (LINE PARAMETER ALSO)
                        $X3DBS::.BLKW   1               ; DEFAULT BLOCK SIZE
                        $X3MBS::.BLKW   1               ; MAXIMUM BLOCK SIZE
                        $X3DWS::.BLKW   1               ; DEFAULT WINDOW SIZE
                        $X3MWS::.BLKW   1               ; MAXIMUM WINDOW SIZE
                        $X3CAL::.BLKW   1               ; CALL TIMER
                        $X3CLR::.BLKW   1               ; CLEAR TIMER
                        $X3RES::.BLKW   1               ; RESET TIMER
                        $X3RST::.BLKW   1               ; RESTART TIMER
                        $X3MCL::.BLKW   1               ; MAXIMUM CLEAR VALUE
                        $X3MRE::.BLKW   1               ; MAXIMUM RESETS VALUE
                        $X3MRS::.BLKW   1               ; MAXIMUM RESTARTS VALUE
                        ;
                        ; LINE PARAMETERS
                        ;
                        $SLTA:: .BLKW   1               ; SLT ADDRESS
                        $DLCPV::.BLKW   1               ; DLC PDV ADDRESS
                        $DDMPV::.BLKW   1               ; DDM PDV ADDRESS
                        $SLN::  .BLKW   1               ; SYSTEM LINE NUMBER
                                .BLKB   48.             ; MUX SLN'S
                        $$CSR:: .BLKW   1               ; CSR ADDRESS
                        $$PRI:: .BLKW   1               ; INTERRUPT PRIORITY
                        $$VECT::.BLKW   1               ; VECTOR ADDRESS
                        $$MTP:: .BLKW   1               ; MULTIPOINT STATION COUNT
                        $$SNUM::.BLKB   48.             ; MULTIPOINT STATION NUMBERS
                        $$APR:: .BLKB   48.             ; MULTIPOINT STATION ACTIVE POLLING RATIO
                        $$DPR:: .BLKW   1               ; DEAD POLLING RATIO
                        $$MUX:: .BLKW   1               ; MUX UNIT COUNT
                        $$SCSR::.BLKW   16.             ; SECONDARY CSR TABLE
                        $$DCHA::.BLKW   2               ; DEVICE CHARACTERISTICS
                                .BLKW   48.*2           ; MUX CHARACTERISTICS
```



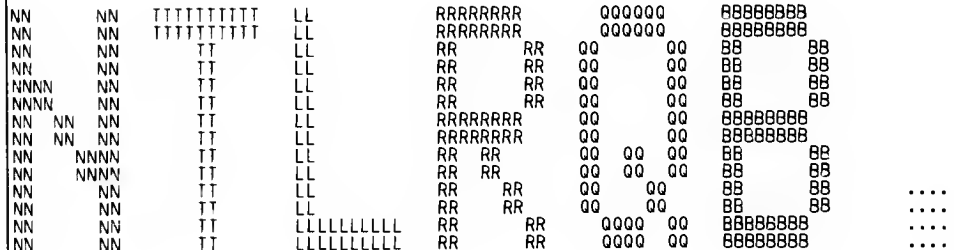
```
AIC
ALOCB
ALOCT
CB.CCB=
CB.DDM=
CB.DLC=
CB.RDB=
CB.SDB=
CB.SLI=
CB.XLB=
CCBAF
CCBAL
CCBCT
CCBLH
CCBNM
CCBSZ
CC.LLC=
CEAVL
CEPWR
CE.ABO=
CE.DAO=
CE.DIS=
CE.ERR=
CE.ILN=
CE.LTO=
CE.MOP=
CE.NTE=
CE.RTE=
CE.SRC=
CE.STP=
CE.TME=
CE.TMO=
CE.UNS=
CF.CHN=
CF.EOM=
CF.HDR=
CF.LB =
CF.LIN=
CF.SOM=
CF.SYN=
CF.TRN=
CLKHD
CMFRK
CM.CIR=
CM.FMT=
CM.HRD=
CM.LIN=
CM.LOO=
CM.XLO=
CP.DCF=
CP.HDL=
CP.PS
CP.PSI
CP.XCF
CP.2FR=
CRAVL
CS.ABO
```

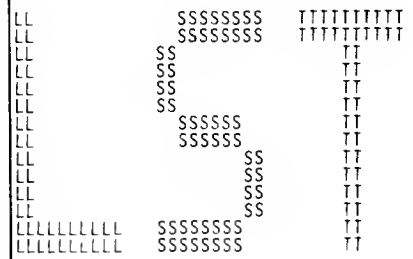
Symbol table

```
AIC     000474R          CS.BRO= 000002          DEAC1   000050RG  002 FS.RST= 000000          LN.REF= 000002
ALOCB   000266RG  002    CS.BUF= 000200          DECPT   000150RG  002 FS.RTN= 001000          LN.SER= 000002
ALOC1   000264RG  002    CS.CES= 000002          DEUMR   000014RG  002 FS.SET= 005000          LN.STA= 000017
CB.CCB= 000002           CS.CHN= 000010          DEVHD   000036RG  002 FS.SFC= 005000          LN.SUB= 000360
CB.DDM= 000040           CS.CMP= 000200          EXRQN   000016RG  002 FS.SFR= 006000          LN.TRI= 000006
CB.DLC= 000020           CS.DCR= 000400          EXSIZ   000024RG  002 FS.SFS= 004000          LOOP    000004R
CB.RDB= 000004           CS.DEF= 000004          FC.CCP= 000020            FS.SPW= 040000          LTMFC   000120RG  002
CB.SDB= 000010           CS.DEV= 000002          FC.CTL= 000006            FS.STM= 000000          L.COST  000015
CB.SLI= 000100           CS.DIS= 000040          FC.KCP= 000016            FS.STP= 002000          L.CTL   000012
CB.XLB= 000001           CS.ENA= 000001          FC.KIL= 000004            FS.STR= 001000          L.CVA   177776
CCBAF   000122RG  002    CS.ENB= 000020          FC.MAN= 000024            FS.TRM= 003000          L.DDM   000002
CCBAL   000132RG  002    CS.ERR= 100000          FC.MLD= 000026            FS.WLB= 001000          L.DDS   000004
CCBCT   000114RG  002    CS.FTL= 001000          FC.PCT= 000030            FS.XKL= 002000          L.DLC   000003
CCBLH   000220RG  002    CS.HCR= 000001          FC.PWR= 000022            FS.XOF= 010000          L.DLM   000006
CCBNM   000116RG  002    CS.HFE= 002000          FC.RCE= 000002            FS.XON= 007000          L.DLS   000010
CCBSZ   000222RG  002    CS.LST= 040000          FC.RCP= 000014            FS.ZER= 002000          L.FLG   000000
CC.LLC= 000200           CS.MTL= 004000          FC.TIM= 000010            FTAB    001236R   003 L.KRBA  000016
CEAVL   000240RG  002    CS.RNG= 000010          FC.XCP= 000012            HEADR   000270RG  002 L.LEN = 000022
CEPWR   000106RG  002    CS.ROV= 000004          FC.XME= 000000            KISAR0= ****** GX       L.MPF   000022
CE.ABO= 100362           CS.RSN= 010000          FILHD   000152RG  002 KISAR5= ****** GX       L.NMST  000020
CE.DAO= 100346           CS.SHU= 000001          FL.CEX= 000010 G          KISAR6= ****** GX       L.NSTA  000014
CE.DIS= 100366           CS.SID= 000002          FL.CHA= 000020 G          KISAR7= ****** GX       L.OWNR  000021
CE.ERR= 100370           CS.STR= 000004          FL.DDM= 000001 G          KSAR0   000062RG  002 L.UNT   000013
CE.ILN= 100350           CS.SUC= 000001          FL.DLC= 000002 G          KSAR5   000006RG  002 MAXOV   000102RG  002
CE.LTO= 100356           CS.TMO= 020000          FL.ECH= 000040 G          KSAR6   000010RG  002 MS.BUF= 000100 G
CE.MOP= 100372           CS.XUR= 000004          FL.FDX= 000100 G          KSAR7   000012RG  002 MS.CIR= 010000 G
CE.NTE= 100361           CXOPT   000230RG  002   FL.HDX= 000200 G          LDBAF   000126RG  002 MS.CSR= 000001 G
CE.RTE= 100376           C.ADD   000034          FL.KMX= 000400 G          LDPWF   000052RG  002 MS.EXT= 002000 G
CE.SRC= 100364           C.BID   000003          FL.LLC= 000004 G          LF.ACT= 100000          MS.INC= 001000 G
CE.STP= 100352           C.BLF   000014          FL.LMC= 001000 G          LF.BRO= 000400          MS.LIN= 004037 G
CE.TME= 100354           C.BUF   000014          FL.MUX= 002000 G          LF.BWT= 000007          MS.LLC= 000040 G
CE.TMO= 100374           C.BUF2  000024          FL.NTI= 004000 G          LF.ENA= 002000          MS.MTP= 000010 G
CE.UNS= 100344           C.CNT   000020          FL.PX1= 010000 G          LF.LPB= 001000          MS.MUX= 004020 G
CF.CHN= 000001           C.CNT1  000020          FL.PX2= 020000 G          LF.MDC= 000100          MS.NT1= 000300 G
CF.EOM= 000004           C.CNT2  000030          FL.TOP= 040000 G          LF.MFL= 000400          MS.NT2= 000400 G
CF.HDR= 000020           C.FLG   000022          FL.WRN= 100000 G          LF.MTP= 000020          MS.PAR= 000200 G
CF.LB = 100000           C.FLG1  000022          FMASK   000020RG  002 LF.PAC= 000200          MS.PRI= 000002 G
CF.LIN= 000002           C.FLG2  000032          FS.AST= 000000            LF.RDY= 040000          MS.SEC= 004000 G
CF.SOM= 000010           C.FNC   000010          FS.CIB= 002000            LF.REA= 010000          MS.VCT= 000004 G
CF.SYN= 000040           C.LIN   000006          FS.CRA= 001000            LF.SER= 000040          NBIAS   000100RG  002
CF.TRN= 000100           C.LNK   000000          FS.DIS= 013000            LF.TIM= 000010          NETPF   000202RG  002
CLKHD   000022RG  002    C.MOD   000011          FS.DVC= 001000            LF.UNL= 020000          NONSI   000046RG  002
CMFRK   000066RG  002    C.NSP   000004          FS.ENB= 012000            LF.X2P= 000000          NSO     000156RG  002
CM.CIR= 000002           C.PRO   000042          FS.EXI= 001000            LGDDB   000206RG  002 NS1     000160RG  002
CM.FMT= 100000           C.RSV   000002          FS.GET= 005000            LGPDV   000204RG  002 NS2     000162RG  002
CM.HRD= 000002           C.STA   000007          FS.HLT= 000000            LGSTT   000154RG  002 NS3     000164RG  002
CM.LIN= 000000           C.STS   000012          FS.INI= 000000            LLCAL   000200RG  002 NS4     000166RG  002
CM.LOO= 000001           C.URM   177776          FS.KIL= 000000            LLCTA   000136RG  002 NS5     000170RG  002
CM.XLO= 000004           C.XACP  000004          FS.LCL= 100000            LN.CLO= 000000          NS6     000172RG  002
CP.DCF= 000040           C.XID   000035          FS.LTM= 001000            LN.DUM= 000005          NS7     000174RG  002
CP.HDL= 000007           C.XLEN  000044          FS.MNT= 004000            LN.LOA= 000004          NTLPT   000232RG  002
CP.PS = 177400           C.XPLI  000040          FS.MSN= 014000            LN.LOO= 000003          NTUIC   000074RG  002
CP.PSI= 000200           C.XPT   000034          FS.REA= 001000            LN.OAU= 000003          OBJHD   000234RG  002
CP.XCF= 000100           C.XSVC  000042          FS.RET= 000000            LN.OFF= 000001          PARHD   000054RG  002
CP.2FR= 000030           C.XTC   000037          FS.REZ= 003000            LN.ON = 000000          PAVL    000244RG  002
CRAVL   000026RG  002    C.X25   000036          FS.RLB= 002000            LN.OOP= 000004          PDVID   000176RG  002
CS.ABO= 000100           DEACB   000040RG  002   FS.RNG= 011000            LN.OPE= 000001          PDVNM   000142RG  002
```


Symbol table

FILEID**NTLRQB


NN NN TTTTTTTTTT LL RRRRRRRR QQQQQQ BBBBBBBB
NN NN TTTTTTTTTT LL RRRRRRRR QQQQQQ BBBBBBBB
NN NN TT LL RR RR QQ QQ BB BB
NN NN TT LL RR RR QQ QQ BB BB
NNNN NN TT LL RR RR QQ QQ BB BB
NNNN NN TT LL RR RR QQ QQ BB BB
NN NN NN TT LL RRRRRRRR QQ QQ BBBBBBBB
NN NN NN TT LL RRRRRRRR QQ QQ BBBBBBBB
NN NNNN TT LL RR RR QQ QQ QQ BB BB
NN NNNN TT LL RR RR QQ QQ QQ BB BB
NN NN TT LL RR RR QQ QQ BB BB ....
NN NN TT LL RR RR QQ QQ BB BB ....
NN NN TT LLLLLLLLLL RR RR QQQQ QQ BBBBBBBB ....
NN NN TT LLLLLLLLLL RR RR QQQQ QQ BBBBBBBB ....



NTLRQB - PARSE REQUEST BLOCK MACRO V05.03b Saturday 29-Jun-85 00:32
Table of contents

```
                                        .SBTTL  SPRO - CHECK SET PROCESS REQUEST BLOCK

                                ;+
                                ;
                                ; SPRO - CHECK ''SET PROCESS'' REQUEST BLOCK
                                ;
                                ; INPUTS:
                                ;       $PDVA=PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R4 CLEARED IF NO WORK TO DO
                                ;
                                ;-

000404  012700  000146'         SPRO:   MOV     #PBITS,R0       ; DO BIT TWIDDLING
000410                                  CALL    $BITS           ;   ...
000414  032712  000000G                 BIT     #FL.LLC,(R2)    ; IS THIS AN LLC?
000420  001403                         BEQ     10$             ; IF EQ, NO
000422  052767  000000G 000000G         BIS     #MS.LLC,$MISS   ; INDICATE LLC INFO MISSING
000430  016700  000000G         10$:    MOV     $PDVA,R0        ; GET PDV ADDRESS

000434  016067  000000G 000000G         MOV     Z.PCB(R0),$PCB  ; IS PROCESS CORE-RESIDENT? / GET PCB IF IT IS
000442  001407                         BEQ     30$             ; IF EQ, NO
000444  032712  000000G                 BIT     #FL.LLC,(R2)    ; IS THIS AN LLC?
000450  001403                         BEQ     20$             ; IF EQ, NO
000452  005760  000000G                 TST     Z.DAT(R0)       ; IS LINE TABLE PRESENT? (CLEARS C-BIT)
000456  001401                         BEQ     30$             ; IF EQ, NO
000460  005004                 20$:    CLR     R4              ; NO WORK TO DO!
000462                         30$:    RETURN
```

```
                                                        CMP     #140000,(R1)+   ; IS CHANNEL 0 ASSIGNED?
                                                        BEQ     40$             ; IF EQ, NO
                                                        TST     -(R1)           ; RESET CHANNEL POINTER
                                                20$:    CMP     #100000,(R1)+   ; IS CHANNEL 1-N ASSIGNED?
                                                        BEQ     40$             ; IF EQ, NO
                                                        MOV     @SLTMA,-(SP)    ; SAVE ADDRESS OF SLT TABLE

                                                        MOV     #30$,-(SP)      ; ALLOW FOR CO-ROUTINE CLEANUP
                                                        JSR     R2,$SAVVR       ; SAVE R0,R1,R2
                                                        MOV     -(R1),R1        ; GET SYSTEM LINE NUMBER
                                                        BIC     #^C<377>,R1     ; ..AS BYTE
                                                        ASL     R1              ; CONVERT TO WORD INDEX
                                                        ADD     R1,10.(SP)      ; ADJUST SLT POINTER
                                                        MOV     @10.(SP),10.(SP) ; POINT TO SLT
                                                        ADD     #L.FLG,10.(SP)  ; POINT AT FLAGS WORD OF SLT
                                                        RETURN                  ; RESTORE R0,R1,R2

                                                30$:    BIT     #LF.ACT,@(SP)+  ; IS CHANNEL 1-N ACTIVE?
                                                        BEQ     40$             ; IF EQ, NO
                                                        EMSG$   Z3              ; CHANNEL STILL ACTIVE (SETS R0 NON-ZERO)
                                                40$:    CMP     R1,R2           ; SCAN ENTIRE CHANNEL TABLE
                                                        BNE     20$             ; ....
                                                50$:    NEG     R0              ; ANY ERRORS?
                                                        BCS     190$            ; IF CS, YES

                                                        .IF DF  R$$11M

                                                        CLR     R1              ; INDICATE NO DEVICE ERROR
                                                        SWSTK$  110$            ;* ENTER KERNEL MODE
                                                        MOV     DEVHD,R2        ;* AND RSX11M DEVICE POINTER
                                                60$:    MOV     (R2),R2         ;* GET NEXT DEVICE IN LIST
                                                        BEQ     100$            ;* IF EQ, ALL DONE
                                                        CMP     Z.PCB(R4),D.PCB(R2) ;* IS OUR LLC ALSO THIS DEVICE?
                                                        BNE     60$             ;* IF NE, NO
                                                        MOV     D.UCB(R2),R3    ;* GET FIRST UCB ADDRESS
                                                        MOVB    D.UNIT(R2),R4   ;* AND FIRST UNIT NUMBER
                                                70$:    BIT     #DV.MNT,U.CW1(R3) ;* MOUNTABLE DEVICE?
                                                        BNE     80$             ;* IF NE, YES
                                                        BITB    #US.BSY,U.STS(R3) ;* IS UNIT BUSY?
                                                        BEQ     90$             ;* IF EQ, NO
                                                        MOV     D.NAM(R2),4(SP) ;* SAVE NAME OF BUSY DEVICE IN R1
                                                        BR      100$            ;* AND RETURN TO USER MODE
                                                80$:    BITB    #US.MNT,U.STS(R3) ;* IS UNIT MOUNTED?
                                                        BNE     90$             ;* IF NE, NO
                                                        MOV     D.NAM(R2),2(SP) ;* SAVE NAME OF MOUNTED DEVICE IN R0
                                                        BR      100$            ;* AND RETURN TO USER MODE
                                                90$:    ADD     D.UCBL(R2),R3   ;* GET NEXT UCB ADDRESS
                                                        INCB    R4              ;* AND NEXT UNIT NUMBER
                                                        CMPB    D.UNIT+1(R2),R4 ;* ALL UNITS CHECKED?
                                                        BHIS    70$             ;* IF HIS, NO
                                                100$:   RETURN                  ;* RETURN TO USER MODE

                                                110$:   TST     R1              ; WERE ANY UNITS BUSY?
                                                        BEQ     120$            ; IF EQ, NO
                                                        EMSG$R  Z7              ; DEVICE BUSY
                                                120$:   MOV     R0,R1           ; ANY UNITS MOUNTED?
                                                        BEQ     170$            ; IF EQ, NO
```

```
NTLRQB - PARSE REQUEST BLOCK    MACRO V05.03b Saturday 29-Jun-85 00:32  Page 30-2
Symbol table

$SAVRG= ****** GX       $$DEV = ****** GX       $$MTP = ****** GX       $$VECT= ****** GX       .KMCPF  000102R     002
$SAVVR= ****** GX       $$EXT = ****** GX       $$MUX = ****** GX       .CBIAS= ****** GX       .SCEX   000020R     002
$SLN  = ****** GX       $$HTIM= ****** GX       $$NODC= ****** GX       .CCEX   000054R     002 .SCIR   000110R     002
$SLTA = ****** GX       $$INC = ****** GX       $$PAR = ****** GX       .CLIN   000072R     002 .SLIN   000040R     002
$TIOUT= ****** GX       $$LINX= ****** GX       $$PRI = ****** GX       .CPRO   000062R     002 .SPRO   000026R     002
$TRIB = ****** GX       $$LRB = ****** GX       $$SNUM= ****** GX       .CXCSR= ****** GX       ..SCSR= ****** GX
$$CSR = ****** GX       $$LTIM= ****** GX       $$TBL = ****** GX       .HOME = ****** GX

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        003724    001   (RW,I,LCL,REL,CON)
DATA    001522    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13932 Words  ( 55 Pages)
Size of core pool: 15496 Words  ( 59 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:41.11
SY:NTLRQB.V2,[132,134]NTLRQB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLRQB
```



```
NTLRQB     CREATED BY  MACRO  ON 29-JUN-85 AT 00:33     PAGE 1
```

```
                        ;+
                        ; $AZFS - SETUP ASCIZ FILE SPEC
                        ;
                        ; INPUTS:
                        ;       R0=ASCIZ UIC ADDRESS (0=USE DEFAULT)
                        ;       R1=ASCIZ FILE NAME ADDRESS (0=USE DEFAULT)
                        ;       R2=ASCIZ EXTENSION ADDRESS (0=USE DEFAULT)
                        ;
                        ; OUTPUTS:
                        ;       R1=ASCIZ FILE SPEC ADDRESS
                        ;       R2=FILE SPEC LENGTH (EXCLUDING ZERO BYTE)
                        ;-
000042                  $AZFS::
000042  010246                 MOV     R2,-(SP)        ; SAVE EXTENSION ADDRESS
000044  012702  000243'         MOV     #FILE,R2        ; PLACE WHERE FILE SPEC WILL BE ASSEMBLED
000050  005700                 TST     R0              ; IS THERE A UIC ADDRESS?
000052  001002                 BNE     10$             ; IF NE, YES
000054  012700  000032'         MOV     #$DFUIC,R0      ; ELSE USE DEFAULT UIC
000060  112022         10$:    MOVB    (R0)+,(R2)+     ; COPY ASCIZ UIC INTO FILE SPEC
000062  001376                 BNE     10$             ;  ...
000064  005302                 DEC     R2              ; DISCARD ZERO BYTE
000066  010100                 MOV     R1,R0           ; COPY FILE NAME ADDRESS
000070  001002                 BNE     15$             ; IF NE, THERE IS ONE
000072  012700  000112'         MOV     #$DFNAM,R0      ; ELSE USE DEFAULT NAME
000076  010001         15$:    MOV     R0,R1           ; SAVE FILE NAME ADDRESS FOR LATER
000100  112022         20$:    MOVB    (R0)+,(R2)+     ; COPY ASCIZ FILE NAME INTO FILE SPEC
000102  001376                 BNE     20$             ;  ...
000104  112762  000056  177777 MOVB    #'.,-1(R2)      ; DEPOSIT "."
000112  012600                 MOV     (SP)+,R0        ; RESTORE EXTENSION ADDRESS
000114  001002                 BNE     30$             ; IF NE, THERE IS ONE
000116  012700  000121'         MOV     #DFEXT,R0       ; ELSE USE DEFAULT EXTENSION
000122  112022         30$:    MOVB    (R0)+,(R2)+     ; COPY ASCIZ EXTENSION INTO FILE SPEC
000124  001376                 BNE     30$             ;  ...
000126  005302                 DEC     R2              ; DISCARD ZERO BYTE
000130  010267  000030'         MOV     R2,FNPTR        ; SAVE POINTER FOR POSSIBLE ERROR MESSAGE
000134  012700  000001         MOV     #1,R0           ; SETUP LOCAL FILE NAME POINTER
000140                          CALL    SLFNP           ; FOR NEXT GCL LEVEL
000144  112120         40$:    MOVB    (R1)+,(R0)+     ; COPY FILE NAME INTO LOCAL AREA
000146  001376                 BNE     40$             ;  ...
000150  012701  000243'         MOV     #FILE,R1        ; SET FILE SPEC ADDRESS
000154  160102                 SUB     R1,R2           ; SET FILE SPEC LENGTH
000156                          RETURN
```

```
                                ; CHANNEL TABLE FLAGS
                                ;
            100000              ZS.ASN==100000  ; CHANNEL IS FREE TO ASSIGN
            140000              ZS.BSY==140000  ; CHANNEL IS IN PROCESS OF ASSIGNMENT

                                ;***
                                ; LOCAL SYMBOLS
                                ;***


                                ;
                                ; TEMPLATE PROCESSING FUNCTION CODES
                                ;
                                ; WARNING - DON'T CHANGE THESE CODES WITHOUT CHANGING TMPEX2, TMPEX3,
                                ;           TMPEX5, TMPEX6, AND TMPEX7
                                ;
                                ;
000020                                  .ASECT
            000001                      .=1                     ; 0=END OF FILE
000001                          .ADDRB::.BLKB   1
000002                          .ADDRW::.BLKB   1
000003                          .BLKB:: .BLKB   1
000004                          .BLKW:: .BLKB   1
000005                          .BYTE:: .BLKB   1
000006                          .CORE:: .BLKB   1
000007                          .CSR::  .BLKB   1
000010                          .CTIM:: .BLKB   1
000011                          .DVCHA::.BLKB   1
000012                          .EOF::  .BLKB   1
000013                          .FILE:: .BLKB   1
000014                          .INT::  .BLKB   1
000015                          .LIBR:: .BLKB   1
000016                          .LINKS::.BLKB   1
000017                          .LFILE::.BLKB   1
000020                          .LSTHD::.BLKB   1
000021                          .LTAB:: .BLKB   1
000022                          .MPTAB::.BLKB   1
000023                          .MXTAB::.BLKB   1
000024                          .NODE:: .BLKB   1
000025                          .PRI::  .BLKB   1
000026                          .PRTOV::.BLKB   1
000027                          .SCOM:: .BLKB   1
000030                          .SECSR::.BLKB   1
000031                          .SLNB:: .BLKB   1
000032                          .SLNW:: .BLKB   1
000033                          .STNB:: .BLKB   1
000034                          .STNW:: .BLKB   1
000035                          .UNB::  .BLKB   1
000036                          .UNW::  .BLKB   1
000037                          .WORD:: .BLKB   1
000040                          .APR::  .BLKB   1
000041                          .DPRB:: .BLKB   1
000042                          .DPRW:: .BLKB   1
000043                          .TIME:: .BLKB   1
000044                          .MXPTB::.BLKB   1
000045                          .PECHA::.BLKB   1
000046                          .X3C01::.BLKB   1
000047                          .X3C02::.BLKB   1
```

```
                                ;+
                                ; $TMP - TEMPLATE FILE PROCESSOR
                                ;
                                ; THIS ROUTINE IS CALLED TO PROCESS THE TEMPLATE FILE AND TO LOAD THE
                                ; DATA BASE INTO CORE.
                                ;
                                ; INPUTS:
                                ;       $BIAS=PROCESS RELOCATION BIAS
                                ;       $$CSR=CSR ADDRESS
                                ;       $$PRI=INTERRUPT PRIORITY
                                ;       $$VECT=INTERRUPT VECTOR
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000000                          $TMP::
000000  016767  000000G 000012'        MOV     $SLTA,$SLTSV    ; SAVE SLT ADDRESS
000006                                  CALL    $TM.IN          ; INITIALIZE COUNTERS, OPEN PRIMARY TEMPLATE
000012  103416                         BCS     10$             ; IF CS, OPEN FAILURE OR NOTHING TO DO
000014                                  CALL    $TPARS          ; LOAD TPARS TABLES
000020                                  CALL    $TMFIL          ; PROCESS TEMPLATE FILE
000024                                  CALL    $TMCHK          ; CHECK FOR DDM INTERRUPT LINKAGE ERROR
000030                                  CALL    $TMALT          ; ALLOCATE DATA BASE
000034                                  CALL    $TMEXP          ; EXPAND BINARY INFORMATION
000040                                  CALL    $TMVCT          ; SETUP DDM VECTOR(S)
000044                                  CALL    $TMLTA          ; STORE LINE TABLE ADDRESS
000050  006167  000000'          10$:   ROL     $FLAG1          ; INTO THE SIGN BIT
000054                                  RETURN
                                ;
                                ; PROCESS TEMPLATE FILE
                                ;
000056                          $TMFIL: CALL    $SAVAL          ; SAVE ALL REGISTERS
000062  016746  000002'                 MOV     $FLAG2,-(SP)    ; SAVE $FLAG2
000066  062767  000002  000032'         ADD     #2,$SZPTR       ; ADJUST SIZE POINTER FOR NEW DEPTH
000074  016777  000014' 000032'         MOV     $SIZE,@$SZPTR   ; SAVE CURRENT VALUE OF $SIZE
000102  005067  000002'          10$:   CLR     $FLAG2          ; CLEAR LINE FLAGS
000106  005067  000076'                 CLR     $VALUE          ; CLEAR EXPRESSION VALUE
000112                                  CALL    TM.RD           ; READ 1 LINE FROM TEMPLATE FILE
000116  103426                         BCS     30$             ; IF CS, END OF FILE
000120  032767  000000G 000000G        BIT     #FL.ECH,$FLAGS  ; IS ECHO ENABLED?
000126  001402                         BEQ     15$             ; IF EQ, NO
000130                                  CALL    $ELINE          ; PRINT TEMPLATE LINE
000134  005001                  15$:   CLR     R1              ; PARSE THE INPUT LINE
000136  012702  000000G                MOV     #$TMPKW,R2
000142  012705  000000G                MOV     #$TMPST,R5
000146                                  CALL    .TPARS
000152  103002                         BCC     20$             ; IF CC, NO PROBLEM
000154                                  CALL    ERR$9           ; SYNTAX ERROR
000160  032767  000002  000002' 20$:   BIT     #F2.END,$FLAG2  ; ".END" SEEN YET?
000166  001745                         BEQ     10$             ; IF EQ, NO - GET NEXT LINE
000170                                  CALL    $CLDEQ          ; CLOSE TEMPLATE FILE
000174  012667  000002'          30$:   MOV     (SP)+,$FLAG2    ; RESTORE $FLAG2
000200  162767  000002  000032'         SUB     #2,$SZPTR       ; RESET SIZE TABLE POINTER
000206                                  RETURN
```

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$TERR	000646 RG	5-189 5-190 5-191 5-192 5-193 #15-486
$TIOUT	= ****** GX	9-331 15-509
$TMALT	= ****** GX	6-231
$TMCHK	= ****** GX	6-230
$TMEXP	= ****** GX	6-232
$TMFIL	000056 R	6-229 #6-241 8-306
$TMFIX	000306 RG	#8-305
$TMLTA	= ****** GX	6-234
$TMP	000000 RG	#6-224
$TMPKW	= ****** GX	6-253
$TMPST	= ****** GX	6-254
$TMVCT	= ****** GX	6-233
$TM.IN	= ****** GX	6-226
$TPARS	= ****** GX	6-228 8-314
$TWARN	000622 RG	#15-478 15-485
$TWE	000636 RG	#15-483
$UBIAS	000050 RG	#5-122
$UDA	000052 RG	#5-124
$VALUE	000076 RG	#5-145 *6-246
$VECT1	000062 RG	#5-132
$VECT2	000064 RG	#5-133
$VECT3	000066 RG	#5-134
$VFDDM	000522 RG	#12-412
$VFDLC	000542 RG	#13-427
$VFLVT	000574 RG	#14-451
$VFSPC	000562 RG	#14-447
.ABFI	000316 RG	#8-312
.BYTE	= ****** GX	11-398
.TPARS	= ****** GX	6-255

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 4

Table of contents

9- 211   PUTAST - PUT COMPLETION AST IN CALLER'S AST QUEUE

6-
7-
8-
9-
10-
12-
13-
14-
18-
19-
20-
21-

Table of contents

6- 67 KMC DATA BASE DESCRIPTION
7- 96 LOCAL DATA AND DEFINITIONS
8- 147 ERROR PROCESSING DATA AND FILE BUFFER
9- 218 KMC LOADER (PROCESS LOAD)
10- 251 KMC LOADER (POWER FAIL)
12- 388 kMC LOAD LIST HOUSEKEEPER
13- 525 KMC MICRO-CODE LOADER
14- 576 LOAD - LOAD MICROCODE
18- 726 MAIN ERROR PROCESSING
19- 769 FILE ERROR SUB-FORMATTER
20- 820 RAM ERROR PREFIX SUB-FORMATTER
21- 860 RAM ERROR SUFFIX SUB-FORMATTER

```
                                        .ENDC
000472                                  CALL    $MAPX                   ;* MAP TO THE NEW BLOCK
000476  010203                          MOV     R2,R3                   ;* COPY VIRTUAL ADDRESS
                                        .IFF
                                        MOV     R0,R3                   ; COPY THE BLOCK'S ADDRESS
                                        .ENDC

000500  005023                          CLR     (R3)+                   ;* NO NEXT BLOCK
000502  016723  000000'                 MOV     KMCNAM,(R3)+            ;* STORE DEVICE NAME
000506  016723  000002'                 MOV     KMCCSR,(R3)+            ;* STORE CSR ADDRESS
000512  016723  000004'                 MOV     KMCNUM,(R3)+            ;* STORE CONTROLLER NUMBER
000516  005023                          CLR     (R3)+                   ;* NO LOCAL MICRO-CODE ADDRESS
000520  005023                  65$:    CLR     (R3)+                   ;* ZERO THE REMAINDER OF THE BLOCK
000522  032703  000077                  BIT     #77,R3                  ;*
000526  001374                          BNE     65$                     ;*

                                        .IF DF  M$$MGE
000530  016711  000012'                 MOV     @MCMP1,(R1)             ;* MAP TO THE LINK WORD
                                        .ENDC

000534  010077  000014'                 MOV     R0,@KMCPN1              ;* POINT PREVIOUS BLOCK AT THE NEW ONE
000540                          70$:    RETURN
```



```
NTLKMC

SYMBOL

SYMBOL

LN.SER
LN.STA
LN.SUB
LN.TRI
LOAD
L$BFLG
L$BLDZ
L$BMXZ
L.COST
L.CTL
L.CVA
L.DDM
L.DDS
L.DLC
L.DLM
L.DLS
L.FLG
L.KRBA
L.LEN
L.MPF
L.NMST
L.NSTA
L.OWNR
L.UNT
M$$MGE

R$$MPL
R$$11M
SF.ACT
SF.ENA
SF.LPE
SF.MFL
SF.PAC
SF.REA
SF.SER
SF.SVC
SF.UNL
S$$BAS

S.COS
S.FLG
S.LEN
S.NMS
S.OWNR
TS$NHE
ZF.DIA
Z.FLG
$ALN18
$AZFS
$CBDSC
$CBOMC
$CLDEC
```

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
LN.SER	= 000002		#5-60								
LN.STA	= 000017		#5-60								
LN.SUB	= 000360		#5-60								
LN.TRI	= 000006		#5-60								
LOAD	000624	R	13-570	13-573	#14-590						
L$BFLG	= ******	GX	15-615								
L$BLDZ	= ******	GX	15-621								
L$BMXZ	= ******	GX	15-622								
L.COST	000015		#5-60								
L.CTL	000012		#5-60	12-414							
L.CVA	177776		#5-60								
L.DDM	000002		#5-60								
L.DDS	000004		#5-60	9-238							
L.DLC	000003		#5-60								
L.DLM	000006		#5-60								
L.DLS	000010		#5-60								
L.FLG	000000		#5-60								
L.KRBA	000016		#5-60								
L.LEN	= 000022		#5-60								
L.MPF	000022		#5-60								
L.NMST	000020		#5-60								
L.NSTA	000014		#5-60								
L.OWNR	000021		#5-60								
L.UNT	000013		#5-60								
M$$MGE	= 000000		7-135	7-144	7-145	10-271	12-418	12-436	12-465	12-472	12-499
			12-518								
R$$MPL	= ******		5-62	7-103	8-184	8-197	11-325	11-381	13-546	17-716	
R$$11M	= 000000		10-275	11-315	12-419	12-473	12-500	15-614			
SF.ACT	= 000200		#5-60								
SF.ENA	= 000100		#5-60								
SF.LPB	= 000004		#5-60								
SF.MFL	= 000040		#5-60								
SF.PAC	= 000020		#5-60								
SF.REA	= 000010		#5-60								
SF.SER	= 000001		#5-60								
SF.SVC	= 000002		#5-60								
SF.UNL	= 000040		#5-60								
S$$BAS	= ******		8-176	8-176	8-177	8-177	8-178	8-178	8-179	8-179	8-180
			8-180	8-181	8-181	8-182	8-182	8-183	8-183		
S.COST	000001		#5-60								
S.FLG	000000		#5-60								
S.LEN	000004		#5-60								
S.NMST	000002		#5-60								
S.OWNR	000003		#5-60								
TS$NHD	= ******	GX	15-615								
ZF.DIA	= ******	GX	13-566								
Z.FLG	= ******	GX	13-566								
$ALN18	= ******	GX	12-486								
$AZFS	= ******	GX	14-594								
$CBDSG	= ******	GX	19-813								
$CBOMG	= ******	GX	20-855	21-887	21-891						
$CLDEQ	= ******	GX	16-674	17-710							

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

```
                                        .SBTTL  $CTSYM - FIND SYMBOLS FROM CETAB
                                ;+
                                ; *** - $CTSYM - FIND SYMBOLS FROM CETAB
                                ;
                                ; INPUT:
                                ;       $NAME = RAD50 PROCESS NAME
                                ;
                                ; OUTPUT:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       $CTBAD = CTB ADDRESS FOR THIS PROCESS (ZERO IF NOT A DDM)
                                ;
                                ;-
                                $CTSYM::
                                        .IF DF  R$$MPL
                                        MOV     $NAME,CTBSYM+2  ; FORM SYMBOL NAME TO LOOK FOR "$CTXXX"
                                        MOV     #$CXUIC,R0      ; GET THE COMM EXEC UIC
                                        MOV     #CETNAM,R1      ; FORM ASCIZ FILE SPEC "[XX,5X]CETAB.STB"
                                        MOV     #STBEXT,R2      ;       ...
                                        CALL    $AZFS           ;       ...
                                        MOV     #CTBLK,R0       ; GET SYMBOL BLOCK TO LOOK FOR
                                        CALL    $SYMBL          ; LOOK FOR IT
                                        BCS     STERR           ; IF CS, FILE ERROR
                                        TST     CTBLK+2         ; WAS SYMBOL DEFINED ?
                                        BEQ     40$             ; IF EQ, NO .. ASSUME IT IS NOT A DDM

                                        SWSTK$  30$             ;* ENTER KERNEL MODE
                                        MOV     @NTLPT,R0       ;* GET NTL HOME BLOCK BIAS
                                        CALL    $MAPX           ;* MAP TO NTL HOME BLOCK
                                        MOV     .CXALL(R2),R1   ;* GET THE CETAB ALLOCATION ADDRESS
                                        CALL    @(SP)+          ;* RESTORE PREVIOUS MAPPING
                                        ADD     CTVAL,R1        ;* CALCULATE CTB ADDRESS
                                        MOV     R1,$CTBAD       ;* SAVE CTB ADDRESS
                                ;
                                ; SEE IF CTB ALREADY IN LIST
                                ;
                                        MOV     CTLST,R0        ;* GET START OF CTB LIST
                                3$:     MOV     (R0),R0         ;* GET NEXT CTB
                                        BEQ     5$              ;* IF EQ, NOT IN LIST
                                        CMP     R0,R1           ;* IS THIS OUR CTB ?
                                        BNE     3$              ;* IF NE, NO .. KEEP LOOKING
                                        MOV     @HEADR,R0       ;* GET USER HEADER ADDRESS
                                        MOV     (R0),R0         ;* GET USER MODE SP
                                        INC     6(R0)           ;* SET USER MODE C-BIT
                                        BR      20$             ;* AND LEAVE
                                5$:     MOV     @CTLST,L.LNK(R1) ;* LINK CTB INTO FRONT OF CTB CHAIN
                                        MOV     R1,@CTLST       ;*      ...
                                ;
                                ; SEARCH DEVICE LIST FOR RE-CONFIGURATION DEVICE
                                ;
                                        MOV     DEVHD,R0        ;* GET START OF DEVICE LIST
                                10$:    MOV     (R0),R0         ;* GET NEXT DCB ADDRESS
                                        BEQ     20$             ;* IF EQ, END OF LIST
                                        CMP     $RCDEV,D.NAM(R0) ;* IS THIS THE RE-CONFIGURATION DEVICE ?
                                        BNE     10$             ;* IF NE, NO .. KEEP LOOKING
                                        MOV     R0,L.DCB(R1)    ;* STORE DCB ADDRESS IN CTB
                                20$:    MOV     R0,2(SP)        ;* RETURN DCB ADDRESS IN USER R0
```

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
SF.UNL	= 000040	#5-65
SQBLK	000070 R	6-193 #6-200 8-354
STBERR	000756 R	10-532 #10-583
STBEXT	000006 R	#6-113 10-528
S.COST	000001	#5-65
S.FLG	000000	#5-65
S.LEN	000004	#5-65
S.NMST	000002	#5-65
S.OWNR	000003	#5-65
TEMP	000000 R	#6-103 *8-344 *8-345 8-346 *8-347 *8-348 8-349 *8-350 *8-351 8-352 *8-356 8-357 *8-358 8-359 *8-363 *8-364 8-365 *8-366 *8-367 8-368
TSKERR	000226 R	7-285 #7-322
T$FLAG	000044	#5-62
T$LIF	000013	#5-62
T$LIFL	000013	#5-62
T$LIFO	000013	#5-62
T$LIFS	000013	#5-62
T$LIN	000000	#5-62
T$LIPS	000006	#5-62
T$LLD	000012	#5-62
T$LLDC	000045	#5-62
T$LLDL	000012	#5-62
T$LLDO	000012	#5-62
T$LLDS	000012	#5-62
T$LLEN	000046	#5-62
T$LOPR	000002	#5-62
T$LTCL	000024	#5-62
T$LTIM	000026	#5-62
T$LTPR	000014	#5-62
T$LTPS	000020	#5-62
T$NAPL	000004	#5-62
T$NFE	000000	#5-62
T$NLEN	000010	#5-62
T$NNUL	000002	#5-62
T$NOPL	000006	#5-62
T$NRNI	000042	#5-62
T$NRPL	000005	#5-62
T$NRUL	000007	#5-62
T$NVR	000001	#5-62
T$RPRI	000040	#5-62
T$SVC	000034	#5-62
T$T5	000030	#5-62
T$T6	000032	#5-62
T1SCL	= ****** GX	8-350 8-366
T100C	= ****** GX	8-347 8-363
XBLK1	000126 R	6-208 #6-215 10-560
XBLK2	000140 R	6-215 #6-218
XBLK3	000152 R	6-218 #6-221
XPTPDV	000002 R	#6-107
Z.AVL	= ****** GX	*7-298
Z.DAT	= ****** GX	11-598 12-645 13-698

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 4

NTLROO
VECTORE

```
                                $$PCHA::.BLKW   1               ; PROTOCOL EMULATOR CHARACTERISTICS
                                        .BLKW   48.             ; MUX CHARACTERISTICS
                                $$LSAD::.BLKW   1               ; LOCAL PHYSICAL STATION ADDRESS
                                $X2MWS::.BLKW   8.              ; MAXIMUM WINDOW SIZE
                                $X2MBS::.BLKW   8.              ; MAXIMUM BLOCK SIZE
                                $X2MRT::.BLKW   8.              ; MAXIMUM RETRANSMIT COUNT
                                $X2RET::.BLKW   8.              ; RETRANSMIT TIMER
                                $X2HLD::.BLKW   8.              ; HOLDBACK TIMER
000464                          $SMTSZ  == <<.-$PDVA>/2>        ; CALCULATE SIZE OF GLOBAL SYMBOL TABLE (WORDS)
                                ;
                                ; ******* NOTE: THE PRECEEDING DATA STRUCTURES MUST BE CONTIGUOUS ******
                                ;

                                        .IF DF  R$$MPL
                                ;
                                ; VECTORED EXEC DIRECTIVE
                                ;
                                TRNVEC::GIN$    GI.VEC,$VECBG,$VECLN

                                        .ENDC

                                ;
                                ; NTINIT PARAMETERS
                                ;
000446'                         .HOME==$$DCHA+4                 ; HOME BLOCK BIAS
000450'                         .BFPTR==.HOME+2                 ; ACTION ROUTINES (BUF$DF)
000452'                         .CCBNM==.BFPTR+2                ; NUMBER OF CCB'S (BUF$DF)
000454'                         .CCBSZ==.CCBNM+2                ; CCB SIZE (BUF$DF)
000456'                         .RDBNM==.CCBSZ+2                ; NUMBER OF RDB'S (BUF$DF)
000460'                         .RDBSZ==.RDBNM+2                ; RDB SIZE (BUF$DF)
000462'                         .SDBNM==.RDBSZ+2                ; NUMBER OF SDB'S (BUF$DF)
000464'                         .SDBSZ==.SDBNM+2                ; SDB SIZE (BUF$DF)
000466'                         .RDBTH==.SDBSZ+2                ; RDB THRESHOLD (BUF$DF)
000470'                         .MXPRO==.RDBTH+2                ; MAXIMUM PROTOCOL OVERHEAD SIZE (BUF$DF)
000472'                         .NTPSZ==.MXPRO+2                ; NTPOOL SIZE NEEDED (RDB/SDB + FUDGE)
000474'                         .PNAM1==.NTPSZ+2                ; FIRST PARTITION NAME (PAR$DF)
000500'                         .PNAM2==.PNAM1+4                ; SECOND PARTITION NAME (PAR$DF)
000504'                         .NAMBF==.PNAM2+4                ; ASCII TO RAD50 CONVERSION BUFFER
000512'                         .PCB==.NAMBF+6                  ; NTPOOL PCB ADDRESS
000514'                         .NOMAP==.PCB+2                  ; SIZE OF NON-UMR-MAPPED AREA
000516'                         .UMRFL==.NOMAP+2                ; UMR FLAG
000520'                         .UMRBA==.UMRFL+2                ; UMR BLOCK ADDRESS
000522'                         .PUMR==.UMRBA+2                 ; UMR ADDRESS CONSTANT ($PUMR OF CEX)
000524'                         .FUDGE==.PUMR+2                 ; FUDGE FACTOR
000526'                         .ALLOC==.FUDGE+2                ; SINGLE WORD POOL LENGTH

                                ;
                                ; PRIMARY STATUS
                                ;
001232  100000  000000          $FLAGS::.WORD   100000,0        ; ENABLE WARNING MESSAGES

000001                          FL.DDM==1                       ; DDM BEING LOADED
000002                          FL.DLC==2                       ; DLC BEING LOADED
000004                          FL.LLC==4                       ; LLC BEING LOADED
000010                          FL.CEX==10                      ; COMM EXEC BEING LOADED
000020                          FL.CHA==20                      ; DDM CHARACTERISTICS UPDATE
000040                          FL.ECH==40                      ; ECHO THE TEMPLATE
```


NTLROO
Symbol

PDVTA
PLGTH
PSIPT
PUMR
P.LGTH=
P.REL =
QASTT
QBIAS
QSTRT
RDBAF
RDBCT
RDBLH
RDBNM
RDBSZ
RDBTH
SDBAF
SDBCT
SDBLH
SDBNM
SDBSZ
SF.ACT=
SF.ENA=
SF.LPB=
SF.MFL=
SF.PAC=
SF.REA=
SF.SER=
SF.SVC=
SF.UNL=
SLHD
SLTMA
SLTNM
SRSTD
STMFC
SYSNM
S$CEX
S.COST
S.FLG
S.LEN
S.LHD
S.NMST
S.OWNR
TKPS
TKTCB
TK100
TTNS
T1SCL
T100C
T100Q
UISAR0
UISAR7
UISDR7
USAR0
USAR7
USDR7

```
PDVTA    000140RG    002 V$$CTR= ****** GX      $LGDDB= ****** GX      $SDBAF= ****** GX      $$DEV   000100RG  003
PLGTH    000056RG    002 XAVL    000250RG  002 $LGPDV= ****** GX      $SDBCT= ****** GX      $$DPR   000376RG  003
PSIPT    000272RG    002 XBIAS   000246RG  002 $LGSTT= ****** GX      $SDBLH= ****** GX      $$EXT   000030RG  003
PUMR     000236RG    002 ZF.INI= ****** GX       $LLCAL= ****** GX      $SDBNM= ****** GX      $$HTIM  001266RG  003
P.LGTH= ****** GX        ZF.MFL= ****** GX       $LLCTA= ****** GX      $SDBSZ= ****** GX      $$INC   000072RG  003
P.REL = ****** GX        ZF.PSE= ****** GX       $LTMFC= ****** GX      $SLIN   000314R        $$LINX  000104RG  003
QASTT    000034RG    002 ZTIME   000064RG  002 $MAXOV= ****** GX      $SLN    000144RG  003  $$LSAD  001110RG  003
QBIAS    000252RG    002 Z.FLG = ****** GX       $MISS   000014RG  003 $SLTA   000136RG  003  $$LTIM  001426RG  003
QSTRT    000254RG    002 Z.NAM = ****** GX       $MXVCT  000076RG  002 $SLTMA= ****** GX      $$MTP   000234RG  003
RDBAF    000124RG    002 Z.PCB = ****** GX       $NAME   000064RG  003 $SLTNM= ****** GX      $$MUX   000400RG  003
RDBCT    000112RG    002 $ALECL= ****** GX       $NBIAS= ****** GX      $SMTSZ= 000464 G       $$NODC  001262RG  003
RDBLH    000224RG    002 $ALMHC= ****** GX       $NETPF= ****** GX      $SPRD   000070R        $$PAR   000074RG  003
RDBNM    000256RG    002 $ALOCB= ****** GX       $NLCFG= ****** GX      $SRSTD= ****** GX      $$PCHA  000746RG  003
RDBSZ    000216RG    002 $ALOC1= ****** GX       $NLDEV= ****** GX      $STMFC= ****** GX      $$PDIN  001264RG  003
RDBTH    000104RG    002 $ALTLC= ****** GX       $NLGET= ****** GX      $SYSNM= ****** GX      $$PRI   000230RG  003
SDBAF    000130RG    002 $BIAS   000066RG  003 $NLINI= ****** GX      $TKPS = ****** GX      $$PRM   000004RG  003
SDBCT    000110RG    002 $CCBAF= ****** GX       $NLPUT= ****** GX      $TKTCB= ****** GX      $$SCSR  000402RG  003
SDBLH    000226RG    002 $CCBAL= ****** GX       $NLRQB= ****** GX      $TK100= ****** GX      $$SNUM  000236RG  003
SDBNM    000260RG    002 $CCBCT= ****** GX       $NONSI= ****** GX      $TMP  = ****** GX      $$TASK  000006RG  003
SDBSZ    000262RG    002 $CCBLH= ****** GX       $NS0  = ****** GX      $TRIB   001566RG  003  $$TBL   000102RG  003
SF.ACT= 000200           $CCBNM= ****** GX       $NS1  = ****** GX      $TTNS = ****** GX      $$VECT  000232RG  003
SF.ENA= 000100           $CCBSZ= ****** GX       $NS2  = ****** GX      $T1SCL= ****** GX       .ALLOC= 000526RG  003
SF.LPB= 000004           $CCEX = ****** GX       $NS3  = ****** GX      $T100C= ****** GX       .AMAXC  000052RG  003
SF.MFL= 000040           $CEAVL= ****** GX       $NS4  = ****** GX      $T100Q= ****** GX       .AMAXH  000050RG  003
SF.PAC= 000020           $CEPWR= ****** GX       $NS5  = ****** GX      $URM    000022RG  003  .BFPTR= 000450RG  003
SF.REA= 000010           $CLIN = ****** GX       $NS6  = ****** GX      $VECBG  000000RG  002  .CCBNM= 000452RG  003
SF.SER= 000001           $CLKHD= ****** GX       $NS7  = ****** GX      $VECLN= 000136 G        .CCBSZ= 000454RG  003
SF.SVC= 000002           $CMFRK= ****** GX       $NTCFS= ****** GX      $XAVL = ****** GX       .FUDGE= 000524RG  003
SF.UNL= 000040           $CPRO = ****** GX       $NTCF3= ****** GX      $XBIAS= ****** GX       .HOME = 000446RG  003
SLHD     000070RG    002 $CRAVL= ****** GX       $NTICF= ****** GX      $XBUF2  001256RG  003  .MAXC  000034RG  003
SLTMA    000144RG    002 $CTBAD  000020RG  003 $NTL    000000R         $XFER = ****** GX       .MAXH  000036RG  003
SLTNM    000146RG    002 $CTIM   000106RG  003 $NTLPT= ****** GX      $XLEN2  001260RG  003  .MXPRO= 000470RG  003
SRSTD    000032RG    002 $CTSYM= ****** GX       $NTUIC= ****** GX      $XLIN = ****** GX       .NA    000042RG  003
STMFC    000242RG    002 $CXOPT= ****** GX       $OBJHD= ****** GX      $XPRO = ****** GX       .NAMBF= 000504RG  003
SYSNM    000002RG    002 $DDMPV  000142RG  003 $PARHD= ****** GX      $X2HLD  001212RG  003  .NBEA  000046RG  003
S$CEX = ****** GX        $DEACB= ****** GX       $PCB    000070RG  003 $X2MBS  001132RG  003  .NBRA  000044RG  003
S.CDST   000001          $DEAC1= ****** GX       $PDVA   000062RG  003 $X2MRT  001152RG  003  .NN    000032RG  003
S.FLG    000000          $DECPT= ****** GX       $PDVID= ****** GX      $X2MWS  001112RG  003  .NOMAP= 000514RG  003
S.LEN    000004          $DEUMR= ****** GX       $PDVNM= ****** GX      $X2RET  001172RG  003  .NTPSZ= 000472RG  003
S.LHD = ****** GX        $DEVHD= ****** GX       $PDVTA= ****** GX      $X3CAL  000120RG  003  .PCB  = 000512RG  003
S.NMST   000002          $DLCPV  000140RG  003 $PSIPT= ****** GX      $X3CLR  000122RG  003  .PNAM1= 000474RG  003
S.OWNR   000003          $DSINI= ****** GX       $PUMR = ****** GX      $X3DBS  000110RG  003  .PNAM2= 000500RG  003
TKPS     000004RG    002 $EXRQN= ****** GX       $QASTT= ****** GX      $X3DWS  000114RG  003  .PUMR = 000522RG  003
TKTCB    000072RG    002 $EXSIZ= ****** GX       $QBIAS= ****** GX      $X3MBS  000112RG  003  .RDBNM= 000456RG  003
TK100    000134RG    002 $FEATR  000026RG  003 $QSTRT= ****** GX      $X3MCL  000130RG  003  .RDBSZ= 000460RG  003
TTNS     000030RG    002 $FILHD= ****** GX       $RCDEV  000024RG  003 $X3MRE  000132RG  003  .RDBTH= 000466RG  003
T1SCL    000214RG    002 $FLAGS  001232RG  003 $RDBAF= ****** GX      $X3MRS  000134RG  003  .ROUTB  000054RG  003
T100C    000212RG    002 $FMASK= ****** GX       $RDBCT= ****** GX      $X3MWS  000116RG  003  .ROUTM  000040RG  003
T100Q    000210RG    002 $FUNC   000012RG  003 $RDBLH= ****** GX      $X3RES  000124RG  003  .RPRI  000056RG  003
UISARO= ****** GX        $HEADR= ****** GX       $RDBNM= ****** GX      $X3RST  000126RG  003  .SDBNM= 000462RG  003
UISAR7= ****** GX        $ICBI = 000016 G        $RDBSZ= ****** GX      $ZTIME= ****** GX       .SDBSZ= 000464RG  003
UISDR7= ****** GX        $ICBLN  000016RG  003 $RDBTH= ****** GX      $$APR   000316RG  003  .UMRBA= 000520RG  003
USARO    000060RG    002 $INICO= ****** GX       $RSXNM  000000RG  003 $$AST   000002RG  003  .UMRFL= 000516RG  003
USAR7    000042RG    002 $LDBAF= ****** GX       $SCEX   000552R        $$CSR   000226RG  003  .$$$$.= 000034
USDR7    000044RG    002 $LDPWF= ****** GX       $SCIR   000636R        $$DCHA  000442RG  003  ..SCSR  000060RG  003
```


5-
6-
7-
8-
9-
10-
11-
12-
13-
14-
15-
16-
17-
18-
19-
20-
21-
22-
23-
24-
25-
26-
27-
28-
29-
30-

NTLRQB - PARSE REQUEST BLOCK MACRO V05.03b Saturday 29-Jun-85 00:32
Table of contents


```
     5-    54    MACRO DEFINITIONS
     6-    64    LOCAL DATA
     7-   130    ERROR MESSAGES
     8-   196    $NLRQB - PROCESS RECEIVE DATA BLOCK
     9-   243    SCEX - CHECK SET SYS REQUEST BLOCK
    10-   288    CEXLD - CHECK THAT COMM EXEC IS LOADED
    11-   312    PDVA - FIND PDV ADDRESS
    12-   348    CHKPP - CHECK PROCESS PARAMETERS
    13-   391    SPRO - CHECK SET PROCESS REQUEST BLOCK
    14-   421    SLTA - GET SLT ADDRESS
    15-   491    CHKLP - CHECK LINE PARAMETERS
    16-   592    CLRPP - CLEAR PROCESS PARAMETERS
    17-   611    SLIN - CHECK SET LINE REQUEST BLOCK
    18-   633    CPRO - CHECK CLEAR PROCESS REQUEST BLOCK
    19-   781    CLIN - CHECK CLEAR LINE REQUEST BLOCK
    20-   811    KMCPF - CHECK KMC POWERFAIL RELOAD REQUEST BLOCK
    21-   866    CCEX - CHECK CLEAR CEX REQUEST BLOCK
    22-  1010    SCIR - SET CIRCUIT
    23-  1057    $BITS - COPY BITS FROM REQUEST BLOCK TO $FLAGS
    24-  1085    $FPDV - FIND PDV
    25-  1125    $FSLT - FIND SLT
    26-  1189    $RERR - PRINT ERROR MESSAGE
    27-  1229    REP7 - REPLACE REQUESTING TASK'S NAME
    28-  1246    REP7A - REPLACE REQUESTING TASK'S NAME AND CHANNEL NUMBER
    29-  1291    DEV7 - REPLACE REQUESTING TASK'S NAME AND DEVICE NAME
    30-  1320    REP7B - REPLACE REQUESTING TASK'S NAME AND LINE ID
```


NTLRQB - PARSE REQUEST BLOCK MACRO V05.03b Saturday 29-Jun-85 00:32 Page 4


```
NTLRQB
SLTA -
    42
    42
    42
    42
    42
    42
    42
    42
    42
    43
    43
    43
    43
    43
    43
    43
    43
    43
    44
    44
    44
    44
    44
    44
    44
    44
    44
    45
    45
    45
    45
    45
    45
    45
    46
    46
    46
    46
    46
    46
    46
    46
    46
    47
    47
    47
    47
    47
    47
```

```
                                        .SBTTL  SLTA - GET SLT ADDRESS

                                ;+
                                ; SLTA - GET SLT ADDRESS
                                ;
                                ; INPUTS:
                                ;       R3=REQUEST BLOCK POINTER
                                ;               .BLKW   1       ; LR.LIN - LINE NAME
                                ;               .BLKB   1       ; LR.CTL - CONTROLLER NUMBER
                                ;               .BLKB   1       ; LR.UNT - UNIT NUMBER
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       $SLTA=SLT ADDRESS
                                ;       $SLN=SYSTEM LINE NUMBER
                                ;-
                                SLTA:   MOV     (R3)+,R0        ; GET LINE NAME
000464  012300                          BEQ     101$            ; IF ZERO, RETURN FAILURE
000466  001507                          MOV     R0,$NAME        ; SAVE IT
000470  010067  000000G                 CALL    $FPDV           ; LOOK FOR PDV ADDRESS
000474                                  BCS     101$            ; IF CS, NO SUCH PROCESS
000500  103502                          MOV     R0,$PDVA        ; SAVE PDV ADDRESS
000502  010067  000000G                 BITB    #ZF.DDM,Z.FLG(R0) ; IS DDM BIT SET IN PDV?
000506  132760  000000G 000000G        BEQ     101$            ; IF EQ, NO
000514  001474                          MOV     R0,$DDMPV       ; SAVE DDM PDV ADDRESS
000516  010067  000000G                 BITB    #ZF.LMC,Z.FLG(R0) ; MICRO-CODE LOAD NEEDED?
000522  132760  000000G 000000G        BEQ     5$              ; IF EQ, NO
000530  001403                          BIS     #FL.LMC,$FLAGS  ; UPDATE NETLDR STATUS
000532  052767  000000G 000000G 5$:     CLR     -(SP)           ; SET MAXIMUM UNIT COUNT FOR NON-MUX
000540  005046                          BITB    #ZF.MUX,Z.FLG(R0) ; MUX DEVICE?
000542  132760  000000G 000000G        BEQ     10$             ; IF EQ, NO
000550  001416                          BIS     #FL.MUX,$FLAGS  ; UPDATE NETLDR STATUS
000552  052767  000000G 000000G        MOV     #15.,(SP)       ; MAXIMUM OF 16. UNITS/CONTROLLER
000560  012716  000017                  BITB    #ZF.KMX,Z.FLG(R0) ; COMIOP MUX DEVICE?
000564  132760  000000G 000000G        BEQ     10$             ; IF EQ, NO
000572  001405                          BIS     #FL.KMX,$FLAGS  ; UPDATE NETLDR STATUS
000574  052767  000000G 000000G        MOV     #47.,(SP)       ; MAXIMUM OF 48. UNITS/CONTROLLER
000602  012716  000057                  CLR     R1              ; GET CONTROLLER NUMBER
000606  005001                  10$:    BISB    (R3)+,R1        ;
000610  152301                          CLR     R2              ; AND UNIT NUMBER
000612  005002                          BISB    (R3)+,R2        ;
000614  152302                          CMP     (SP)+,R2        ; VALID UNIT NUMBER?
000616  022602                          BLO     111$            ; IF LO, NO
000620  103435                          CALL    $FSLT           ; TRY TO FIND SLT
000622                                  BCS     101$            ; IF CS, NOT FOUND
000626  103427                          MOV     R0,$SLTA        ; SET SLT ADDRESS
000630  010067  000000G                 MOV     R1,$SLN         ; AND SYSTEM LINE NUMBER
000634  010167  000000G                 MOV     R2,$$MUX        ; SAVE UNIT COUNT (FOR $NLCFG)
000640  010267  000000G                 MOVB    L.DLC(R0),R1    ; GET DLC PDV INDEX
000644  116001  000003                  ADD     @PDVTA,R1       ; ADD PDV TABLE ADDRESS
000650  067701  000000G                 MOV     .CBIAS(R1),R2   ; GET DLC PDV ADDRESS
000654  016102  000000G
                                20$:    BITB    #ZF.DLC,Z.FLG(R2) ; IS DLC BIT SET IN PDV?
000660  132762  000000G 000000G        BEQ     121$            ; IF EQ, NO
000666  001415
```

```
                                        EMSG$R  Z4              ; DEVICE STILL MOUNTED
                                        .ENDC
002540  012701  000003          130$:   MOV     #L.DLC,R1       ; ASSUME DLC
002544  132764  000000G 000000G       BITB    #FL.DLC,Z.FLG(R4) ; IS IT A DLC?
002552  001002                          BNE     140$            ; IF NE, YES
002554  012701  000002                  MOV     #L.DDM,R1       ; IT MUST BE A DDM
002560  017702  000000G         140$:   MOV     @SLTMA,R2       ; GET SLT TABLE ADDRESS
002564  017703  000000G                 MOV     @SLTNM,R3       ; AND NUMBER OF SYSTEM LINES
002570  001421                  145$:   BEQ     170$            ; IF ZERO, NONE TO CHECK
002572  012246                  150$:   MOV     (R2)+,-(SP)     ; GET NEXT SLT ON STACK
002574  062716  000000                  ADD     #L.FLG,(SP)     ; POINT AT FLAGS WORD
002600  032736  040000                  BIT     #LF.RDY,@(SP)+  ; IS THIS LINE STILL READY?
002604  001411                          BEQ     160$            ; IF EQ, NO - SKIP TO NEXT LINE
002606  016246  177776                  MOV     -2(R2),-(SP)    ; GET SLT ADDRESS ON STACK
002612  060116                          ADD     R1,(SP)         ; ADD IN INDEX POINTER

002614  123667  000126'                 CMPB    @(SP)+,PDVX     ; DOES THIS LINE REFERENCE OUR PROCESS?
002620  001003                          BNE     160$            ; IF NE, NO
002622                                  EMSG$   Z5              ; LINE STILL READY (SETS R0 NON-ZERO)
002630                          160$:   SOB     R3,150$         ; LOOP <SYSTEM LINE COUNT> TIMES
002634  005267  000126'        170$:   INC     PDVX            ; NEXT PDV
002640  005367  000134'                DEC     PDVNUM          ; ANY MORE PDV ENTRIES
002644  003402                          BLE     180$            ; IF LE, NO
002646  000167  177336                  JMP     10$             ; GET NEXT ENTRY
002652  005400                  180$:   NEG     R0              ; SET C-BIT IF ERROR OCCURRED
002654                          190$:   RETURN
```

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
CCEX	002156 R	6-102 #21-884
CEXLD	000164 R	6-92 6-96 6-101 6-103 6-106 6-109 6-111 #10-302
CHKLP	000730 R	6-98 #15-516
CHKPP	000262 R	6-94 #12-369
CLIN	002054 R	6-108 #19-795
CLRPP	001244 R	6-90 6-99 #16-607
CPRO	001306 R	6-105 #18-653
DEVHD	= ****** GX	18-728 21-952
DEV7	003532 R	7-169 7-172 #29-1309
DV.MNT	= ****** GX	18-735 21-959
D.DSP	= ****** GX	*18-749
D.NAM	= ****** GX	18-739 18-743 21-963 21-967
D.PCB	= ****** GX	18-731 *18-750 21-955
D.UCB	= ****** GX	18-733 21-957
D.UCBL	= ****** GX	18-745 21-969
D.UNIT	= ****** GX	18-734 18-747 21-958 21-971
ERRBUF	000170 R	#7-137 26-1207
ERRMSG	000160 R	#7-135 26-1219
FE.CEX	= ****** GX	9-262 10-303
FL.CEX	= ****** GX	9-260
FL.DDM	= ****** GX	11-337 22-1037
FL.DLC	= ****** GX	11-337 18-761 21-985 22-1037
FL.ECH	= ****** GX	6-126
FL.FDX	= ****** GX	6-124
FL.HDX	= ****** GX	6-125
FL.KMX	= ****** GX	14-459
FL.LLC	= ****** GX	11-337 12-377 13-407 13-414 18-674 21-914 22-1037
FL.LMC	= ****** GX	14-451
FL.MUX	= ****** GX	14-455 25-1172
FL.NTI	= ****** GX	9-280
FL.TOP	= ****** GX	6-127
FMASK	= ****** GX	9-262 10-303
FMAX	= 000010	#6-85 8-224
FMT7	001444 R	7-145 7-146 7-147 7-148 7-149 7-150 7-151 7-152 7-154 7-158 7-159 7-160 7-161 7-167 7-169 7-172 7-175 7-177 7-178 #7-187
FMT7A	001[illegible] R	7-168 #7-188
FMT7B	001511 R	7-176 #7-189
FM.7	= 000000	#7-145 #7-146 #7-147 #7-148 #7-149 #7-150 #7-151 #7-152 #7-154 #7-158 #7-159 #7-160 #7-161 #7-167 #7-169 #7-172 #7-175 #7-177 #7-178
FM.7A	= 000000	#7-168
FM.7B	= 000000	#7-176
FTAB	000000 R	#6-77 6-85 8-228
I$$AS	= ******	15-543 21-887
KMCPF	002120 R	6-110 #20-825
LBITS	000136 R	#6-124 17-625
LF.ACT	= 100000	#5-62 18-716 19-801 21-940
LF.BRO	= 000400	#5-62 18-699
LF.BWT	= 000007	#5-62
LF.ENA	= 002000	#5-62
LF.LPB	= 001000	#5-62

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 2

```
                        ;+
                        ; $REP.C - REPLACE "*" WITH CURRENT LEVEL FILE NAME
                        ; $REP.P - REPLACE "*" WITH PREVIOUS LEVEL FILE NAME
                        ;
                        ; THESE ROUTINES ARE STANDARD ERROR MESSAGE REPLACERS. $REP.C SHOULD BE USED
                        ; WHEN THE FILE IN QUESTION IS STILL OPEN, AND $REP.P SHOULD BE USED WHEN
                        ; THE FILE HAS ALREADY BEEN CLOSED.
                        ;
                        ; INPUTS:
                        ;       R3=FORMAT STRING ADDRESS
                        ;       R4=ERROR MESSAGE STRING ADDRESS
                        ;
                        ; OUTPUTS:
                        ;       R0=DESTROYED
                        ;       R3,R4=UPDATED
                        ;-
                        
                                .ENABL  LSB
                        
000160                  $REP.C::
000160  005000                  CLR     R0              ; USE CURRENT GCL LEVEL
000162  000402                  BR      10$
000164                  $REP.P::
000164  012700  000001          MOV     #1,R0           ; USE PREVIOUS GCL LEVEL
000170  112314          10$:    MOVB    (R3)+,(R4)      ; LOOK FOR THE ASTERISK
000172  122724  000052          CMPB    #'*,(R4)+       ;       ...
000176  001374                  BNE     10$             ;       ...
000200  005304                  DEC     R4              ; SKIP THE ASTERISK
000202                          CALL    SLFNP           ; SETUP LOCAL FILE NAME POINTER
000206  112024          20$:    MOVB    (R0)+,(R4)+     ; COPY ASCIZ FILE NAME
000210  001376                  BNE     20$             ; INTO THE ERROR MESSAGE
000212  005304                  DEC     R4              ; SKIP THE ZERO BYTE
000214                          RETURN
                        
                                .DSABL  LSB
```

```
                                .X3C03::.BLKB   1
                                .X3C04::.BLKB   1
                                .X3C05::.BLKB   1
                                .X3C06::.BLKB   1
                                .X3C07::.BLKB   1
                                .X3C08::.BLKB   1
                                .X3C09::.BLKB   1
                                .X3C10::.BLKB   1
                                .X3C11::.BLKB   1
                                .X2CH1::.BLKB   1
                                .X2CH2::.BLKB   1
                                .X2CH3::.BLKB   1
                                .X2CH4::.BLKB   1
                                .X2CH5::.BLKB   1
                                        .PSECT

                                ;
                                ; DATA BLOCK OFFSETS (POINTED TO BY $NTLPT - NTPOOL SPACE)
                                ;
                                ; WARNING - DON'T CHANGE THESE OFFSETS WITHOUT RE-BUILDING VNP AND NTINIT.
                                ;
                                        .ASECT
                                        .=0
                                        .BLKW   1               ; NTPOOL PCB ADDRESS (SETUP BY NTINIT)
                                        .BLKW   1               ; NTPOOL UMR BLOCK (SETUP BY NTINIT)
                                .CXCSR::.BLKW   1               ; VNP CSR->SYSTEM LINE LIST
                                .CXPCB::.BLKW   1               ; COMM EXEC PCB ADDRESS
                                .CXSYM::.BLKW   3               ; PROCESS VERIFICATION SYMBOLS
                                .CXVSM::.BLKW   3               ; VECTORED PROCESS VERIFICATION SYMBOLS
                                .CXUNL::.BLKW   1               ; FIRST UNLOAD BLOCK ADDRESS
                                .CXVEC::.BLKW   1               ; FIRST KMC CSR BLOCK ADDRESS
                                .CXLB1::.BLKW   6               ; TWO LIBRARY REFERENCES
                                .CXLB2::.BLKW   6               ;   ...
                                .CXLBR::.BLKW   1               ; FIRST EXTRA LIBRARY BLOCK ADDRESS
                                .CXALL::.BLKW   1               ; CETAB ALLOCATION ADDRESS
                                .CXLEN::.BLKW   1               ; CETAB ALLOCATION LENGTH
                                        .BLKW   1               ; AVAILABLE
                                        .PSECT

                                ;
                                ; COMM EXEC ADDRESSING BIAS
                                ;
                        000000  AB==0
                                        .IF DF R$$MPL
                                                                ; RSX-11M+ ICB LENGTH DEFINED AT TASK BUILD
                                $KRBLN == 26                    ; RSX-11M+ KRB LENGTH
                                        .IFF ; DF R$$MPL
                        000004  $KRBLN == 4                     ; RSX-11M/S KRB LENGTH
                                        .ENDC ; DF R$$MPL


                                ;
                                ; LIBRARY BLOCK OFFSETS
                                ;
                                        .ASECT
                                        .=0
                                        .BLKW   3               ; LIBRARY NAME IN RAD50
```

```
                                        ;+
                                        ; TM.RD - READ ONE RECORD FROM TEMPLATE FILE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT=SUCCESS/FAILURE
                                        ;       R0,R1,R2=DESTROYED
                                        ;       R3=RECORD LENGTH
                                        ;       R4=RECORD ADDRESS
                                        ;-
000210                                  TM.RD:  GCL$    #TMPBF,#132.    ; READ NEXT RECORD
000230  016700  000000G                         MOV     $CLIOS,R0       ; GET GCL IOSB ADDRESS
000234  016003  000002                          MOV     2(R0),R3        ; GET RECORD LENGTH
000240  010367  000352'                        MOV     R3,TMPSZ        ; SAVE IT
000244  012704  000146'                        MOV     #TMPBF,R4       ; GET RECORD ADDRESS
000250  105710                                  TSTB    (R0)            ; SUCCESS? (CLEARS C-BIT)
000252  100401                                  BMI     101$            ; IF MI, NO
000254                                          RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000256  122710  000000G                 101$:   CMPB    #IE.EOF,(R0)    ; END OF FILE?
000262  001405                                 BEQ     111$            ; IF EQ, YES
000264  122710  000000G                         CMPB    #IE.RBG,(R0)    ; BUFFER OVERFLOW?
000270  001404                                 BEQ     121$            ; IF EQ, YES
000272                                          CALLR   ERR$10          ; READ FAILURE
000276                                  111$:   CALLR   ERR$11          ; NO END STATEMENT (WARNING)
000302                                  121$:   CALLR   ERR$12          ; READ OVERFLOW
```

NTLTMP CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	6-226	6-228	6-229	6-230	6-231	6-232	6-233	6-234	6-241	6-247
	6-251	6-255	6-257	6-260	7-278	8-306	8-313	9-327	10-371	11-396
	12-415	13-430	15-489	15-496	15-509	15-512				
CALLR	7-294	7-295	7-296	8-307	8-314	9-331				
EMSG$	#5-58	10-371	12-415	13-430						
EMSG$R	#5-58	14-459	14-460							
GCL$	#5-58	7-278								
NTLER$	#5-58	5-189	5-190	5-191	5-192	5-193				
RETURN	6-236	6-263	7-285	10-380	11-401	12-416	13-431	14-454	15-514	

FILEID**NTLUNL

```
                        .TITLE  NTLGET - GET SEND/RECEIVE DATA PACKET
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - SEND/RECEIVE DATA PACKET I/O ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

```
                        .TITLE  NTLKMC - LOAD KMC MICRO-CODE
                        .IDENT  /V05.00/

                        .NLIST  TTM,BEX

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - KMC MICRO-CODE LOADING ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                                        .SBTTL  KMC MICRO-CODE LOADER


                                ;+
                                ; KMCLD - LOAD MICRO-CODE FROM TASK FILE
                                ;
                                ; INPUTS:
                                ;       R0 = PDV/DEVICE NAME IN RAD50 FORMAT
                                ;       R4 = KMC CSR ADDRESS
                                ;       $URM = UNIBUS RUN MASK (RSX-11M+)
                                ;       R0 AND R4 DO NOT HAVE TO BE SAVED
                                ;
                                ; OUTPUTS:
                                ;       KMC RAM LOADED FROM FILE [131,54]XXXMC.TSK (XXX=PDV/DEVICE NAME)
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000542                          KMCLD:
                                ;
                                ; SAVE THE DEVICE NAME AS AN ASCIZ STRING, IN CASE OF RAM ERROR
                                ;
                                        .IF DF  R$$MPL

                                        TSTB    $NCPU                   ;M+ IS THIS A MULTI-PROCESSOR SYSTEM ?
                                        BEQ     5$                      ;M+ IF EQ, NO
                                        MOV     $URM,SETAF+S.AFAF       ;M+ RUN ON THE RIGHT PROCESSOR
                                        DIR$    #SETAF                  ;M+ ...SET THE AFFINITY
                                5$:                                     ;M+
                                        .ENDC

000542  010046                          MOV     R0,-(SP)                ;B-SAVE RAD50 FORM
000544  010001                          MOV     R0,R1                   ;B-PLACE IN R1
000546  012700  000006'                 MOV     #KMCDEV,R0              ;B-PLACE ASCIZ FORM HERE
000552                                  CALL    $C5TA                   ;B-CONVERT RAD50 TO ASCIZ
000556  105010                          CLRB    (R0)                    ;B-MARK END OF STRING
000560  012600                          MOV     (SP)+,R0                ;B-RECOVER RAD50 FORM
                                ;
                                ; USE $AZFS TO SET UP THE PROPER FILE NAME STRING FOR BOTH
                                ; FILE OPENNING AND FOR ERROR MESSAGES
                                ;
000562  016702  000000G                 MOV     $PDVA,R2                ;B-GET PDV ADDRESS
000566  032762  000000G 000000G         BIT     #ZF.DIA,Z.FLG(R2)       ;B-DO WE NEED TO LOAD DIAGNOSTICS?
000574  001406                          BEQ     10$                     ;B-BR IF NO
000576  010046                          MOV     R0,-(SP)                ;B-SAVE RAD50 NAME
000600  012701  015030                  MOV     #^RDG,R1                ;B-R1 = LAST TWO CHARACTERS
000604                                  CALL    LOAD                    ;B-LOAD DIAGNOSTICS
000610  012600                          MOV     (SP)+,R0                ;B-RESTORE RAD50 NAME
000612  012701  050670          10$:    MOV     #^RMC,R1                ;B-R0=FIRST 2 OR 3 CHARS, R1=LAST 2
000616                                  CALL    LOAD                    ;B-LOAD MICROCODE
000622                                  RETURN
```

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES			
$CLIOS	= ****** GX	14-599	17-689	17-699	19-811
$CLOPE	= ****** GX	14-598			
$C5TA	= ****** GX	12-490	13-558		
$DFN2	= ****** GX	14-590			
$ERR3A	000142 R	#8-176	17-683		
$ERR3C	000174 R	#8-177	17-686		
$ERR3D	000230 R	#8-178	17-690		
$ERR3E	000276 R	#8-179	17-693		
$ERR3F	000342 R	#8-180	17-696		
$ERR3H	000404 R	#8-181	17-700		
$ERR3I	000436 R	#8-182	17-708		
$ERR3J	000500 R	#8-183	12-493		
$IOSB	= ****** GX	17-689	17-699		
$KMC	000000 RG	#9-236			
$KMCPF	000036 RG	#10-271			
$LBN	= ****** GX	*16-662	*16-663		
$MAPX	= ****** GX	10-281	10-286	12-423	12-503
$NAME	= ****** GX	9-243	12-412	12-488	
$NTLPT	= ****** GX	10-285	12-422		
$PDVA	= ****** GX	13-565			
$READ	= ****** GX	16-643			
$REP.C	= ****** GX	19-799			
$REP.P	= ****** GX	19-796			
$RLBL	= ****** GX	15-610			
$SLTA	= ****** GX	9-237	12-413		
$TIOUT	= ****** GX	18-766			
$$CSR	= ****** GX	9-244	12-416		
.CXKMC	= ****** GX	10-288	10-290	12-429	

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 4

```
                                        RETURN                  ;* RETURN TO USER MODE
                                        BCS     40$             ; IF CS, CTB ALREADY IN LIST
                                30$:    TST     R0              ; FAILURE ?
                                        BEQ     101$            ; IF EQ, YES
                                        CALL    $HRCIN          ; RE-INITIALIZE HRC...
                                        BCC     40$             ; IF CC, SUCCESS
                                        EMSG$   9Z              ; ELSE, WARN SOMEBODY
                                        .ENDC   ; DF R$$MPL
000456  000241                  40$:    CLC
000460                                  RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
                                        .IF DF  R$$MPL
                                101$:   EMSG$R  9A              ; ERROR - COULD NOT FIND RE-CONFIGURATION DEVICE
                                STERR:  MOV     ETAB1-2(R0),R0  ; GET ERROR MESSAGE CONTROL BLOCK ADRESS
                                        CALLR   @(R0)+          ; PRINT ERROR MESSAGE
                                        .ENDC
```

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES				
Z.DSP	= ******	GX	*7-289	*7-291			
Z.PCB	= ******	GX	7-272				
$ADJ	= ******	GX	11-600				
$ALECL	001220	RG	#13-683				
$ALMHC	001022	RG	#12-629				
$ALN18	= ******	GX	12-642	13-695			
$AZFS	= ******	GX	7-283	10-529			
$BIAS	= ******	GX	*7-290	7-305	8-346	8-349	8-352
$CAT5	= ******	GX	10-480	10-483	10-523		
$CKT	= ******	GX	11-604				
$CTSYM	000456	RG	#9-388				
$DFN	= ******	GX	7-276				
$DFNAM	= ******	GX	10-464				
$ERRQX	= ******	GX	7-321				
$ERR44	= ******	GX	6-228				
$ERR45	= ******	GX	6-229				
$ERR46	= ******	GX	6-230				
$ERR47	= ******	GX	6-231				
$ERR49	= ******	GX	10-581				
$ERR50	= ******	GX	10-582				
$ERR51	= ******	GX	6-236				
$ERR53	= ******	GX	6-238				
$ERR54	= ******	GX	6-239				
$ERR55	= ******	GX	6-240				
$ERR56	= ******	GX	6-241				
$ERR57	= ******	GX	6-242				
$ERR58	= ******	GX	6-243				
$ERR59	= ******	GX	6-244				
$ERR60	= ******	GX	6-245				
$ERR61	= ******	GX	6-246				
$ERR62	= ******	GX	6-247				
$ERR63	= ******	GX	6-248				
$ERR64	= ******	GX	6-249				
$ERR65	= ******	GX	6-250				
$ERR66	= ******	GX	6-251				
$ERR67	= ******	GX	6-252				
$ERR69	= ******	GX	12-664				
$ERR70	= ******	GX	13-727				
$INICO	000764	RG	#11-597				
$LDSIZ	= ******	GX	7-292				
$MAPX	= ******	GX	7-306	10-548	13-703		
$MUL	= ******	GX	13-689				
$NAME	= ******	GX	7-275	8-340			
$NICAH	= ******	GX	11-599				
$PCB	= ******	GX	*7-288				
$PDVA	= ******	GX	7-271	7-286	11-597	12-644	13-697
$PRCLN	000054	RG	#6-188				
$PREXT	000044	RG	6-180	#6-186			
$PRINC	000032	RG	6-155	#6-180			
$PRIO	= ******	GX	7-284				
$PRLLN	000042	RG	#6-182				
$ROL	= ******	GX	11-602				

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 5

```
                000100          FL.FDX==100                     ; FULL-DUPLEX
                000200          FL.HDX==200                     ; HALF-DUPLEX
                000400          FL.KMX==400                     ; COMIOP MULTIPLEXED LINE
                001000          FL.LMC==1000                    ; MICRO-CODE LOAD NEEDED
                002000          FL.MUX==2000                    ; MULTIPLEXED LINE
                004000          FL.NTI==4000                    ; PERFORM NTINIT AS PART OF SET CEX
                010000          FL.PX1==10000                   ; SUCCESSFUL PROCESS I/O TRANSFER (11M)
                020000          FL.PX2==20000                   ;   ...
                040000          FL.TOP==40000                   ; LOAD PROCESS AT TOP OF PARTITION
                100000          FL.WRN==100000                  ; WARNING MESSAGE OUTPUT CONTROL
                                                                ; *** FL.WRN MUST BE THE SIGN BIT ***


                                ;
                                ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                                ;
001236  000552'         FTAB:  .WORD   $SCEX                    ; 1 - SET CEX
001240  000070'                .WORD   $SPRO                    ; 2 - SET PROCESS
001242  000314'                .WORD   $SLIN                    ; 3 - SET LINE
001244  000000G                .WORD   $CCEX                    ; 4 - CLEAR CEX
001246  000000G                .WORD   $CPRO                    ; 5 - CLEAR PROCESS
001250  000000G                .WORD   $CLIN                    ; 6 - CLEAR LINE
001252  000000                 .WORD   0                        ; 7 - NOT IN USE
001254  000636'                .WORD   $SCIR                    ; 10 - SET CIRCUIT


                                ;
                                ; END OF TASK BUFFER POINTERS
                                ;
001256                  $XBUF2::.BLKW   1                       ; ADDRESS OF BINARY BUFFER
001260                  $XLEN2::.BLKW   1                       ; LENGTH OF BINARY BUFFER


                                ;
001262                  $$NODC::.BLKW   1                       ; MAXIMUM NUMBER OF NODE COUNTERS
001264                  $$PDIN::.BLKW   1                       ; FUNCTION CODE FOR PROCESS INITIALIZATION


                                ;
                                ; MAXIMUM VECTOR IN SYSTEM
                                ;
                                .IIF DF V$$CTR, .ERROR          ; THE HIGHEST VECTOR ADDRESS MUST NOT BE
                                                                ; DEFINED AT ASSEMBLY TIME


001266                  $$HTIM::.BLKW   48.                     ; HELLO TIMER
001426                  $$LTIM::.BLKW   48.                     ; LISTEN TIMER
001566                  $TRIB:: .BLKB   1                       ; STATION NUMBER
                                .EVEN

000000                          .PSECT
```

```
NTLROO - NTL ROOT SEGMENT       MACRO V05.03b Saturday 29-Jun-85 00:32  Page 14-3
Symbol table

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        000746    001   (RW,I,LCL,REL,CON)
$$$VEC  000274    002   (RW,D,LCL,REL,CON)
DATA    001570    003   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 15165 Words  ( 60 Pages)
Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:20.17
SY:NTLROO.V2,[132,134]NTLROO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[132,10]NTLROO
```



```
NTLROO     CREATED BY  MACRO  ON 29-JUN-85 AT 00:32     PAGE 1
```

```
NTLRQB
```

```
                        .TITLE  NTLRQB - PARSE REQUEST BLOCK
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - SEND DATA PACKET PARSING ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

```
NTLRQB - PARSE REQUEST BLOCK    MACRO V05.03b Saturday 29-Jun-85 00:32  Page 14-1
SLTA - GET SLT ADDRESS

    478 000670  026702  000000G                 CMP     $DDMPV,R2       ; COMBINATION DDM/DLC?
    479 000674  001001                          BNE     30$             ; IF NE, NO
    480 000676  005002                          CLR     R2              ; INDICATE NO DLC PDV
    481 000700  010267  000000G         30$:    MOV     R2,$DLCPV       ; SAVE DLC PDV ADDRESS
    482 000704                                  RETURN
    483
    484                                 ;
    485                                 ; ERROR CONDITIONS
    486                                 ;
    487 000706                          101$:   EMSG$R  95              ; LINE NOT IN SYSTEM
    488 000714                          111$:   EMSG$R  90              ; ILLEGAL UNIT NUMBER
    489 000722                          121$:   EMSG$R  88              ; GARBAGE IN DLC PDV
```



```
NTLRQB - PARSE REQUEST BLOCK    MACRO V05.03b Saturday 29-Jun-85 00:32  Page 15
CHKLP - CHECK LINE PARAMETERS
```

```
                                        .SBTTL  SCIR - SET CIRCUIT

                                ;+
                                ; SCIR - SET CIRCUIT
                                ;
                                ; INPUTS:
                                ;       R3 - REQUEST BLOCK POINTER
                                ;               .BLKB   10.     ; LR.CIR - CIRCUIT ID
                                ;               .BLKB   1       ; LO.CTL - CONTROLLER NUMBER (DLM ONLY)
                                ;               .BLKB   1       ; LO.STA - STATION NUMBER (DLM ONLY)
                                ;
                                ; OUTPUTS:
                                ;       $SLTA - SLT ADDRESS
                                ;
                                ;-
002656  012700  000146'         SCIR:   MOV     #PBITS,R0       ; DO BIT TWIDDLING
002662                                  CALL    $BITS           ;   ...
002666  032767  004000  000004G         BIT     #LS.DLM,$$LRB+4 ; DATA LINK MAPPING CIRCUIT?
002674  001444                          BEQ     10$             ; BR IF NO
002676  011300                          MOV     (R3),R0         ; GET PROCESS NAME
002700  001443                          BEQ     101$            ; BR IF ERROR
002702  010067  000000G                 MOV     R0,$NAME        ; SAVE IT
002706                                  CALL    $FPDV           ; FIND PDV ADDRESS
002712  103436                          BCS     101$            ; BR IF NOT FOUND
002714  010067  000000G                 MOV     R0,$PDVA        ; SAVE PDV ADDRESS
002720  016001  000000G                 MOV     Z.FLG(R0),R1    ; GET PROCESS FLAGS
002724  042701  000000C                 BIC     #^C<FL.DDM!FL.DLC!FL.LLC>,R1
002730  050167  000000G                 BIS     R1,$FLAGS       ; SET PROCESS TYPE FLAGS
002734  062703  000012                  ADD     #<LO.CTL-LR.CIR>,R3 ; POINT TO CONTOLLER NUMBER
002740  112301                          MOVB    (R3)+,R1        ; GET CONTROLLER NUMBER
002742  005002                          CLR     R2              ; NO UNIT NUMBER
002744                                  CALL    $FSLT           ; FIND SLT ADDRESS
002750  103417                          BCS     101$            ; BR IF ERROR
002752  010067  000000G                 MOV     R0,$SLTA        ; SAVE SLT ADDRESS
002756  010167  000000G                 MOV     R1,$SLN         ; SAVE SYSTEM LINE NUMBER
002762  005067  000000G                 CLR     $$MUX           ; SAVE UNIT COUNT
002766  126013  000014                  CMPB    L.NSTA(R0),(R3) ; LEGAL STATION NUMBER?
002772  103406                          BLO     101$            ; BR IF NO
002774  111367  000000G                 MOVB    (R3),$TRIB      ; SAVE STATION NUMBER
003000  012767  000000G  000000G         MOV     #MS.MTP,$MISS   ; INDICATE MULTIPOINT PARAMTERS MISSING
003006                          10$:    RETURN
                                ;
                                ; ERRORS
                                ;
003010                          101$:   EMSG$R  86              ; CIRCUIT NOT IN SYSTEM
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
LF.MDC	= 000100	#5-62
LF.MFL	= 004000	#5-62
LF.MTP	= 000020	#5-62
LF.PAC	= 000200	#5-62
LF.RDY	= 040000	#5-62 17-628 18-770 19-797 21-994
LF.REA	= 010000	#5-62
LF.SER	= 000040	#5-62
LF.TIM	= 000010	#5-62
LF.UNL	= 020000	#5-62 19-803
LF.X2P	= 000000	#5-62
LN.CLO	= 000000	#5-62
LN.DUM	= 000005	#5-62
LN.LOA	= 000004	#5-62
LN.LOO	= 000003	#5-62
LN.OAU	= 000003	#5-62
LN.OFF	= 000001	#5-62
LN.ON	= 000000	#5-62
LN.OOP	= 000004	#5-62
LN.OPE	= 000001	#5-62
LN.REF	= 000002	#5-62
LN.SER	= 000002	#5-62
LN.STA	= 000017	#5-62
LN.SUB	= 000360	#5-62
LN.TRI	= 000006	#5-62
LO.CON	000006	#5-61
LO.CSR	000006	#5-61
LO.CTL	000014	#5-61 22-1039
LO.LIN	000004	#5-61
LO.PAR	000010	#5-61
LO.PRI	000010	#5-61
LO.SCR	000014	#5-61
LO.STA	000015	#5-61
LO.VCT	000012	#5-61
LR.AST	000026	#5-61
LR.CIR	000002	#5-61 22-1039
LR.CTL	000004	#5-61
LR.EFN	000004	#5-61
LR.LIN	000002	#5-61
LR.PRM	000030	#5-61
LR.PRO	000002	#5-61
LR.STS	000000	#5-61
LR.TCB	000002	#5-61
LR.UNT	000005	#5-61
LS.AST	= 100000	#5-61
LS.CEX	= 000001	#5-61
LS.CIR	= 000010	#5-61
LS.CXO	= 010000	#5-61
LS.CX5	= 000400	#5-61
LS.DLM	= 004000	#5-61 22-1028
LS.ECH	= 001000	#5-61 6-126
LS.FDX	= 004000	#5-61 6-124
LS.HDX	= 010000	#5-61 6-125

```
                                ;+
                                ; $REP.N - REPLACE "*" WITH @$CLIOS (ERROR CODE) CONVERTED TO DECIMAL
                                ; $CUR.N - REPLACE CURRENT LEVEL FILE NAME AND ERROR CODE
                                ; $PRV.N - REPLACE PREVIOUS LEVEL FILE NAME AND ERROR CODE
                                ;
                                ; INPUTS:
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE STRING ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       R3,R4=UPDATED
                                ;-
000216                          $CUR.N::
000216                                  CALL    $REP.C          ; REPLACE CURRENT LEVEL FILE NAME
000222  000402                          BR      $REP.N
000224                          $PRV.N::
000224                                  CALL    $REP.P          ; REPLACE PREVIOUS LEVEL FILE NAME
000230                          $REP.N::
000230  010500                          MOV     R5,R0           ; COPY ERROR MESSAGE BLOCK ADDRESS
000232  005720                          TST     (R0)+           ; NOW POINTS AT TEXT STRING
000234  122720  000050          10$:    CMPB    #'(,(R0)+       ; LOOK FOR LEFT PAREN
000240  001375                          BNE     10$             ;       ...
000242  117701  000000G                 MOVB    @$CLIOS,R1      ; GET ERROR CODE
000246  005002                          CLR     R2              ; SUPPRESS LEADING ZEROES
000250                                  CALL    $CBDSG          ; CONVERT TO SIGNED DECIMAL
000254  112720  000056                  MOVB    #'.,(R0)+       ; DEPOSIT DECIMAL POINT
000260  112720  000051                  MOVB    #'),(R0)+       ; RIGHT PAREN
000264  105010                          CLRB    (R0)            ; AND ZERO BYTE (END OF TEXT INDICATOR)
000266                                  RETURN
```

```
   174 000006                       LB.ADR::.BLKW   1               ; ADDRESS/BIAS
   175 000010                       LB.SIZ::.BLKW   1               ; SIZE
   176 000012                       LB.USE::.BLKW   1               ; USE COUNT
   177          000013              LB.APR==LB.USE+1                ; STARTING APR (MAPPED ONLY)
   178 000000                               .PSECT
   179
   180
   181          000001                      .END
```


Symbol table

```
                                ;
                                ; *** BOTH OF THE FOLLOWING ARE OVERLAY HACKS ***
                                ;
                                ;
                                ; PROCESS TEMPLATE FILE FROM "TMPAC2" OVERLAY
                                ;
000306                          $TMFIX::
000306                                  CALL    $TMFIL          ; PROCESS SECONDARY TEMPLATE
000312                                  CALLR   $FILEX          ; RETURN TO "$FILE" SUBROUTINE
                                ;
                                ; ACCESS BINARY FILE AND RELOAD TPARS TABLES
                                ;
000316                          .ABFI::
000316                                  CALL    $ABFI           ; PROCESS FILE NAME OPERAND
000322                                  CALLR   $TPARS          ; RELOAD TPARS TABLES
```

```
**FILE**ID**NTLUNL
```

```
NN       NN   TTTTTTTTTT   LL           UU       UU   NN       NN   LL
NN       NN   TTTTTTTTTT   LL           UU       UU   NN       NN   LL
NN       NN       TT       LL           UU       UU   NN       NN   LL
NN       NN       TT       LL           UU       UU   NN       NN   LL
NNNN     NN       TT       LL           UU       UU   NNNN     NN   LL
NNNN     NN       TT       LL           UU       UU   NNNN     NN   LL
NN  NN   NN       TT       LL           UU       UU   NN  NN   NN   LL
NN  NN   NN       TT       LL           UU       UU   NN  NN   NN   LL
NN     NNNN       TT       LL           UU       UU   NN     NNNN   LL
NN     NNNN       TT       LL           UU       UU   NN     NNNN   LL
NN       NN       TT       LL           UU       UU   NN       NN   LL                ....
NN       NN       TT       LL           UU       UU   NN       NN   LL                ....
NN       NN       TT       LLLLLLLLLL   UUUUUUUUUU   NN       NN   LLLLLLLLLL        ....
NN       NN       TT       LLLLLLLLLL   UUUUUUUUUU   NN       NN   LLLLLLLLLL        ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL           SS                 TT
LL           SS                 TT
LL           SS                 TT
LL           SS                 TT
LL             SSSSSS           TT
LL             SSSSSS           TT
LL                   SS         TT
LL                   SS         TT
LL                   SS         TT
LL                   SS         TT
LLLLLLLLLL   SSSSSSSS           TT
LLLLLLLLLL   SSSSSSSS           TT
```


NTLUNL - NTL UNLOAD ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:34

Table of contents

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  DIR$,EMSG$,NTLER$,RCVX$,SDAT$,EXIT$S,QIOW$,RETC,NTLDF$
000000                  NTLDF$                  ; DEFINE NETLDR BLOCK OFFSETS
                                .IF DF  R$$11D!I$$AS
                                .MCALL  CRAW$S
                                .IFF
                                .MCALL  RSXOK$
                                .ENDC
                        ;***
                        ; LOCAL DATA
                        ;***
000000                          .PSECT  DATA,D
                        ;
                        ; RECEIVE DATA FROM REQUESTING TASK OR EXIT (DPB)
                        ;
000000                  RCVDAT: RCVX$   ,$$LRB
                        ;
                        ; SEND DATA BACK TO REQUESTING TASK (DPB)
                        ;
000010                  ANSWER: SDAT$   XXXXXX,$$LRB+4
                        ;
                        ; LOAD REQUEST BLOCK
                        ;
000022                  $$LRB:: .BLKW   2               ; REQUESTING TASK'S NAME
000026                          .BLKW   15.
                        ;
                        ; ERROR MESSAGES
                        ;
                                .ENABL  LC
                                .IF DF  R$$11M
000064                          NTLER$  ,40,,$RERR,RTSPC,,<Incompatible With RSX-11M System>
                                .IFF
                                NTLER$  ,40,,$RERR,RTSPC,,<Failed to create Address Window>
                                NTLER$  ,4A,,$RERR,RTSPC,,<PLAS support required for DECnet>
                                .ENDC
                                .DSABL  LC
                        ;
                        ; ERROR MESSAGE FORMAT STRING
                        ;
000136  000            FMTO:    .ASCIZ  '"'
                                .EVEN
```

```
                                ;
                                .MCALL  EMSG$,NTLER$,SLTDF$
                                SLTDF$                  ; DEFINE SLT SYMBOLS AND OFFSETS
                                .IF DF  R$$MPL
                                .MCALL  STAF$,RMAF$S,DIR$
                                .ENDC
```

```
                                        .SBTTL  LOAD - LOAD MICROCODE

                                ;+
                                ;
                                ; LOAD - LOAD MICRCOCODE
                                ;
                                ; INPUTS:
                                ;       R0 - FIRST THREE RAD50 CHARACTERS FOR FILE NAME TO OPEN
                                ;       R1 - LAST TWO RAD50 CHARACTERS FOR FILE NAME TO OPEN
                                ;
                                ; OUTPUTS:
                                ;       KMC RAM LOADED FROM SPECIFIED FILE NAME
                                ;
                                ;-
000624                          LOAD:   CALL    $DFN2                   ;B-PLACE "XXXMC" IN $DFNAM FOR $AZFS
000630  005000                         CLR     R0                      ;B-USE DEFAULT UIC STRING
000632  005001                         CLR     R1                      ;B-NAME ALEADY IN $DFNAM (DEFAULT)
000634  005002                         CLR     R2                      ;B-USE DEFAULT EXTENSION
000636                                  CALL    $AZFS                   ;B-LEAVE R1->STRING, R2=LENGTH
                                ;
                                ; OPEN FILE FOR PROCESSING
                                ;
000642                                  CALL    $CLOPE                  ;B-OPEN FILE WITH R1->NAME, R2=LENGTH
000646  105777  000000G                 TSTB    @$CLIOS                 ;B-DID THE OPEN SUCEED?
000652  100511                         BMI     30$                     ;B-BR ON OPEN ERROR
```

NTLKMC CREATED BY MACRO ON 29-JUN-85 AT 00:31 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	9-245	9-247	10-276	10-281	10-286	11-349	12-420	12-423	12-486	12-490
	12-493	12-501	12-503	13-558	13-570	13-573	14-590	14-594	14-598	15-610
	16-643	16-674	17-683	17-686	17-690	17-693	17-696	17-700	17-708	17-710
	18-752	18-755	18-762	18-766	19-796	19-799	19-813	20-848	20-853	20-855
	20-856	21-885	21-887	21-888	21-891					
EMSG$	#5-58	12-493	17-683	17-686	17-690	17-693	17-696	17-700	17-708	
NTLER$	#5-58	8-176	8-177	8-178	8-179	8-180	8-181	8-182	8-183	
RETURN	9-249	11-316	11-377	12-474	12-523	13-574	17-724	18-758	18-767	19-816
	21-896	21-905								
SLTDF$	#5-58	5-60								
SWSTK$	10-276	12-420	12-501							

FILEID**NTLPRO

```
                                        .SBTTL  PRSYM - LOOK FOR PROCESS SYMBOLS
                                ;+
                                ; PRSYM - LOOK FOR SPECIAL PROCESS SYMBOLS
                                ;
                                ; INPUTS:
                                ;       $DFNAM=ASCIZ PROCESS NAME
                                ;
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       PVAL=DISPATCH TABLE ADDRESS
                                ;-
000462  012700  000000G         PRSYM:  MOV     #$DFNAM,R0      ; FORM DISPATCH TABLE ADDRESS NAME
000466  012701  000013'                 MOV     #DSPTAB+1,R1    ; BY STARTING WITH "$",
000472  012702  000004                  MOV     #4,R2           ; APPENDING THE PROCESS NAME,
000476  005302                  10$:    DEC     R2              ; APPENDING "TB",
000500  112021                          MOVB    (R0)+,(R1)+     ; AND BLANK FILLING OUT TO 6 CHARACTERS.
000502  001375                          BNE     10$             ;       ...

000504  112761  000124  177777        MOVB    #'T,-1(R1)      ;       ...
000512  112721  000102                MOVB    #'B,(R1)+       ;       ...
000516  005702                        TST     R2              ; BLANK FILL IF NEEDED
000520  001404                        BEQ     30$             ;       ...
000522  112721  000040          20$:  MOVB    #' ,(R1)+       ;       ...
000526                                SOB     R2,20$          ;       ...
000532  012700  000012'          30$:  MOV     #DSPTAB,R0      ; NOW TAKE THE ASCII NAME
000536  010701                        MOV     PC,R1           ; AND CONVERT TO RAD50

000540                                  CALL    $CAT5           ; STORING THE RAD50 NAME IN THE GLOBAL
000544  010167  000024'                MOV     R1,PSYM         ; SYMBOL LOOKUP CONTROL BLOCK
000550  010701                        MOV     PC,R1           ;       ...
000552                                  CALL    $CAT5           ;       ...
000556  010167  000026'                MOV     R1,PSYM+2       ;       ...

                                        .IF DF  R$$MPL

                                        MOV     #$DFNAM,R0      ; FORM DISPATCH TABLE ADDRESS NAME
                                        MOV     #VECTAB+1,R1    ; BY STARTING WITH "$",
                                35$:    MOVB    (R0)+,(R1)+     ; COPY NAME
                                        BNE     35$             ;       ...
                                        MOVB    #'V,-1(R1)      ; CREATE $XXXVT
                                        MOVB    #'T,(R1)        ;       ...

                                        MOV     #VECTAB,R0      ; NOW TAKE THE ASCII NAME
                                        MOV     PC,R1           ; AND CONVERT TO RAD50
                                        CALL    $CAT5           ; STORING THE RAD50 NAME IN THE GLOBAL
                                        MOV     R1,VTSYM        ; SYMBOL LOOKUP CONTROL BLOCK
                                        MOV     PC,R1           ;       ...
                                        CALL    $CAT5           ;       ...
                                        MOV     R1,VTSYM+2      ;       ...

                                        MOV     #$DFNAM,R0      ; FORM DISPATCH TABLE ADDRESS NAME
                                        MOV     #VECLEN+1,R1    ; BY STARTING WITH "$",
                                37$:    MOVB    (R0)+,(R1)+     ;       ...
                                        BNE     37$             ;       ...
                                        MOVB    #'V,-1(R1)      ; CREATE $XXXVL
                                        MOVB    #'L,(R1)        ;       ...
```

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES		
$SAVAL	= ****** GX	12-629	13-683	
$SQSRV	000100 R	#6-202	8-357	8-359
$SYMBL	= ****** GX	10-531		
$XFER	000000 RG	#7-270		
$$DEV	= ****** GX	10-519	10-520	
$$NODC	= ****** GX	*13-687	13-700	
$$TBL	= ****** GX	*10-542		
.CXSYM	= ****** GX	10-551		

NTLPRO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 6

NTLROO
$NTL -

43
43
43
44
44
44
44
44
44
44
44
44
45
45
45
45
45
45
45
45
45
45
46
46
46
46
46
46
46
46
46
47
47
47
47
47
47
47
47
47
48
48
48
48
48
48
48

```
                                        .SBTTL  $NTL - NETLDR ENTRY POINT

                                ;+
                                ;
                                ; $NTL - NETLDR ENTRY POINT
                                ;
                                ; THIS PROGRAM IS CALLED TO TRANSFER MCB PROCESSES FROM DISK TO CORE AND
                                ; TO ALLOCATE AND INITIALIZE PROCESS LINE TABLES. NETLDR RECEIVES ITS INPUT
                                ; BY A SEND DATA BLOCK.
                                ;
                                ; INPUTS:
                                ;       DATA BLOCK CONTAINS LOADER DATA.
                                ;
                                ; OUTPUTS:
                                ;       DATA BLOCK (1 WORD OF STATUS) IS RETURNED TO CALLING PROGRAM.
                                ;       CALLING PROGRAM IS RESUMED.
                                ;
                                ;-
000000                          $NTL:
                                        .IF DF  R$$MPL
                                        .IF NDF R$$PRO
                                        DIR$    #TRNVEC         ; SETUP VECTORED EXEC TABLE
                                        BCC     10$             ; IF SUCCESS - BRANCH
                                        IOT

                                10$:    MOV     FMASK,R0        ; GET $FMASK ADDRESS
                                        TST     (R0)+           ; GET $FMASK+2 ADDRESS
                                        BIT     #F2.DAS,(R0)    ; D-SPACE SUPPORTED ?
                                        BEQ     15$             ; IF NO - BRANCH
                                        MOV     #$ICBD,$ICBLN   ; IF YES - SETUP ICB LENGTH

                                        .ENDC
                                        .ENDC

000000  004767  000000G         15$:    CALL    $NLINI          ; INITIALIZE NETLDR
000004  004767  000000G         LOOP:   CALL    $NLGET          ; GET REQUEST BLOCK (OR EXIT)
000010  004767  000000G                 CALL    $NLRQB          ; PROCESS REQUEST BLOCK
000014  103417                         BCS     20$             ; IF CS, BAD INPUT PARAMETER
000016  005767  000012'                 TST     $FUNC           ; IS THIS FUNCTION A NOP?
000022  001417                         BEQ     30$             ; IF EQ, YES
000024  005767  000014'                 TST     $MISS           ; ARE ANY REQUIRED PARAMETERS MISSING?
000030  001403                         BEQ     10$             ; IF EQ, NO
000032  004767  000000G                 CALL    $NLCFG          ; SCAN CONFIGURATION FILE
000036  103406                         BCS     20$             ; IF CS, CONFIG ERROR
000040  016700  000012'         10$:    MOV     $FUNC,R0        ; GET FUNCTION CODE
000044  006300                         ASL     R0              ; MAKE A WORD INDEX
000046  004770  001234'                 CALL    @FTAB-2(R0)     ; TRANSFER TO FUNCTION SPECIFIC ROUTINE
000052  103003                         BCC     30$             ; IF CC, NO PROBLEM
000054  012767  177777  000012' 20$:    MOV     #-1,$FUNC       ; INDICATE ERROR
000062  004767  000000G         30$:    CALL    $NLPUT          ; SEND BACK 1 WORD SUCCESS/FAILURE CODE
000066  000746                         BR      LOOP            ; TRY FOR ANOTHER REQUEST
```


NTLROO

SYMBOL

SYMBOL

AIC
ALOCB
ALOC1
CCBAF
CCBAL
CCBCT
CCBLH
CCBNM
CCBSZ
CEAVL
CEPWR
CLKHD
CMFRK
CRAVL
CXOPT
DEACB
DEAC1
DECPT
DEUMR
DEVHD
EXRQN
EXSIZ
FC.PCT
FILHD
FL.CEX
FL.CHA
FL.DDM
FL.DLC
FL.ECH
FL.FDX
FL.HDX
FL.KMX
FL.LLC
FL.LMC
FL.MUX
FL.NTI
FL.PX1
FL.PX2
FL.TOP
FL.WRN
FMASK
FS.INI
FTAB
HEADR
I$$AS
KISAR0
KISAR5
KISAR6
KISAR7
KSAR0
KSAR5
KSAR6

NTLROO

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
AIC	000474	R	11-672	11-689	#12-734	14-816
ALOCB	000266	RG	#6-211			
ALOC1	000264	RG	#6-210			
CCBAF	000122	RG	#5-151			
CCBAL	000132	RG	#5-155			
CCBCT	000114	RG	#5-148			
CCBLH	000220	RG	#6-188			
CCBNM	000116	RG	#5-149			
CCBSZ	000222	RG	#6-189			
CEAVL	000240	RG	#6-198			
CEPWR	000106	RG	#5-145			
CLKHD	000022	RG	#5-109			
CMFRK	000066	RG	#5-127			
CRAVL	000026	RG	#5-111			
CXQPT	000230	RG	#6-192			
DEACB	000040	RG	#5-116			
DEAC1	000050	RG	#5-120			
DECPT	000150	RG	#6-168			
DEUMR	000014	RG	#5-90			
DEVHD	000036	RG	#5-115			
EXRQN	000016	RG	#5-91			
EXSIZ	000024	RG	#5-110			
FC.PCT =	000030		8-550			
FILHD	000152	RG	#6-169			
FL.CEX =	000010	G	#6-386			
FL.CHA =	000020	G	#6-387			
FL.DDM =	000001	G	#6-383	12-736	12-737	
FL.DLC =	000002	G	#6-384	12-736	12-737	
FL.ECH =	000040	G	#6-388			
FL.FDX =	000100	G	#6-389			
FL.HDX =	000200	G	#6-390			
FL.KMX =	000400	G	#6-391			
FL.LLC =	000004	G	#6-385	8-521		
FL.LMC =	001000	G	#6-392			
FL.MUX =	002000	G	#6-393			
FL.NTI =	004000	G	#6-394			
FL.PX1 =	010000	G	#6-395	8-516	11-679	
FL.PX2 =	020000	G	#6-396	11-696	14-823	
FL.TOP =	040000	G	#6-397			
FL.WRN =	100000	G	#6-398			
FMASK	000020	RG	#5-108			
FS.INI =	000000		8-550			
FTAB	001236	R	#6-404	7-483		
HEADR	000270	RG	#6-212			
I$$AS =	******		12-741	13-781		
KISAR0 =	******	GX	5-125			
KISAR5 =	******	GX	5-87			
KISAR6 =	******	GX	5-88			
KISAR7 =	******	GX	5-89			
KSAR0	000062	RG	#5-125			
KSAR5	000006	RG	#5-87			
KSAR6	000010	RG	#5-88			

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 2

NTLRQB
MACRO [illegible]

NTLRQB
LOCAL

```
                                .SBTTL  MACRO DEFINITIONS

                        ;
                        ; LIBRARY MACROS
                        ;
                                .MCALL  ASL$,EMSG$,EMSG$R,NTLDF$,NTLER$,SLTDF$

000000                          NTLDF$                  ; NETLDR REQUEST BLOCK OFFSETS
000000                          SLTDF$                  ; DEFINE SLT OFFSETS AND SYMBOLS
```

```
                                        .SBTTL  CHKLP - CHECK LINE PARAMETERS

                                ;+
                                ;
                                ; CHKLP - CHECK LINE PARAMETERS
                                ;
                                ; INPUTS:
                                ;       LS.OPT SET IF ANY OPTIONAL PARAMETERS
                                ;       R3=REQUEST BLOCK POINTER
                                ;               .BLKW   1       ; LO.CSR - CSR ADDRESS
                                ;               .BLKW   1       ; LO.PRI - INTERRUPT PRIORITY
                                ;               .BLKW   1       ; LO.VCT - INTERRUPT VECTOR
                                ;               .BLKW   1       ; LO.SCR - SECONDARY CSR
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R3=DESTROYED
                                ;       $$CSR=CSR ADDRESS
                                ;       $$PRI=INTERRUPT PRIORITY
                                ;       $$VECT=INTERRUPT VECTOR
                                ;       ..SCSR=SECONDARY CSR
                                ;       $MISS MAY BE SET NON-ZERO IF ANY FIELDS ARE MISSING
                                ;
                                ;-
000730  012767  000000G 000000G CHKLP:  MOV     #MS.LIN,$MISS   ; ALL PARAMETERS MISSING
000736  032767  000400  000004G         BIT     #LS.OPT,$$LRB+4 ; ANY OPTIONAL PARAMETERS INCLUDED?
000744  001002                         BNE     20$             ; .. YES
000746  012703  000116'                MOV     #NULL,R3        ; NO .. USE NULL INPUT PARAMETERS
000752  012300                 20$:    MOV     (R3)+,R0        ; GET CSR VALUE
000754  001414                         BEQ     30$             ; IF ZERO, FIELD MISSING
000756  022700  160000                 CMP     #160000,R0      ; IS IT AN I/O PAGE ADDRESS?
000762  101111                         BHI     101$            ; IF HI, NO
000764  032700  000003                 BIT     #3,R0           ; IS IT A MULTIPLE OF 4?
000770  001106                         BNE     101$            ; IF NE, NO

                                        .IF DF  R$$MPL

                                        TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
                                        BNE     25$             ; IF NE, YES

                                        .ENDC

000772                                  CALL    $NLCSR          ; IS THE DEVICE ONLINE ?
000776  103514                         BCS     131$            ; .. NO
001000  042767  000000G 000000G 25$:    BIC     #MS.CSR,$MISS   ; INDICATE CSR INCLUDED
001006  010067  000000G         30$:    MOV     R0,$$CSR        ; SET CSR ADDRESS
001012  012300                         MOV     (R3)+,R0        ; GET PRIORITY VALUE
001014  001406                         BEQ     40$             ; IF ZERO, FIELD MISSING
001016  022700  000007                 CMP     #7,R0           ; IS IT RANGE 1-7?
001022  103474                         BLO     111$            ; IF LO, NO
001024  042767  000000G 000000G       BIC     #MS.PRI,$MISS   ; INDICATE PRIORITY INCLUDED
001032                          40$:    ASL$    5,R0            ; CONVERT TO PSW PRIORITY VALUES
001044  010067  000000G                 MOV     R0,$$PRI        ; SET INTERRUPT PRIORITY
001050  012300                         MOV     (R3)+,R0        ; GET VECTOR VALUE
001052  001411                         BEQ     50$             ; IF ZERO, FIELD MISSING
001054  032700  000003                 BIT     #3,R0           ; IS IT A MULTIPLE OF 4?
```

```
                                        .SBTTL  $BITS - COPY BITS FROM REQUEST BLOCK TO $FLAGS

                                ;+
                                ;
                                ; $BITS - COPY BITS FROM REQUEST BLOCK TO $FLAGS
                                ;
                                ; INPUTS:
                                ;       R0 = POINTER INTO MAPPING TABLE
                                ;
                                ; OUTPUTS:
                                ;       R1 = ADDRESS OF REQUEST BLOCK STATUS WORD
                                ;       R2 = ADDRESS OF $FLAGS
                                ;
                                ;-

003016                          $BITS:
003016  012701  000004G                 MOV     #$$LRB+4,R1     ; SETUP ADDRESSES
003022  012702  000000G                 MOV     #$FLAGS,R2      ;       ...
003026  000401                          BR      20$             ; GO TEST FIRST ITEM
003030  005720                  10$:    TST     (R0)+           ; SKIP SECOND WORD IN TABLE ENTRY
003032  005710                  20$:    TST     (R0)            ; END OF TABLE ?
003034  001404                          BEQ     30$             ; IF EQ, YES
003036  032011                          BIT     (R0)+,(R1)      ; IS THE BIT SET IN REQUEST BLOCK ?
003040  001773                          BEQ     10$             ; IF EQ, NO
003042  052012                          BIS     (R0)+,(R2)      ; SET THE BIT IN $FLAGS
003044  000772                          BR      20$
003046                          30$:    RETURN
```

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
LS.LIN	= 000003	#5-61
LS.LMC	= 000007	#5-61
LS.NTI	= 004000	#5-61 9-278
LS.ON	= 000011	#5-61
LS.OPT	= 000400	#5-61 12-370 15-517
LS.PRO	= 000002	#5-61
LS.PWF	= 040000	#5-61
LS.TOP	= 002000	#5-61 6-127
LS.UNF	= 020000	#5-61
LS.X25	= 000012	#5-61
LS.11D	= 004000	#5-61
LX.CEX	= 000004	#5-61
LX.LIN	= 000006	#5-61
LX.PRO	= 000005	#5-61
L.COST	000015	#5-62
L.CTL	000012	#5-62 25-1160 30-1358
L.CVA	177776	#5-62
L.DDM	000002	#5-62 18-763 21-987 25-1158 30-1346
L.DDS	000004	#5-62
L.DLC	000003	#5-62 14-472 18-760 18-763 21-984
L.DLM	000006	#5-62
L.DLS	000010	#5-62
L.FLG	000000	#5-62 17-628 18-696 18-701 18-769 19-796 21-937 21-993
L.KRBA	000016	#5-62
L.LEN	= 000022	#5-62
L.MPF	000022	#5-62 18-701
L.NMST	000020	#5-62
L.NSTA	000014	#5-62 22-1047
L.OWNR	000021	#5-62
L.UNT	000013	#5-62 25-1162 25-1168 30-1364
MS.CSR	= ****** GX	15-536
MS.EXT	= ****** GX	12-369 12-381
MS.INC	= ****** GX	12-369 12-375
MS.LIN	= ****** GX	15-516
MS.LLC	= ****** GX	13-409
MS.MTP	= ****** GX	22-1050
MS.PRI	= ****** GX	15-542
MS.SEC	= ****** GX	15-569
MS.VCT	= ****** GX	15-551
M$$MGE	= 000000	20-826
NTLPT	= ****** GX	20-828
NULL	000116 R	#6-115 12-372 15-519
PBITS	000146 R	#6-126 13-405 22-1026
PDVA	000204 R	6-93 6-104 #11-330
PDVNM	= ****** GX	21-905 24-1105
PDVNUM	000134 R	#6-119 *21-905 *21-1004
PDVTA	= ****** GX	14-473 21-903 24-1103 24-1117 30-1347
PDVX	000126 R	#6-116 18-774 *21-902 21-999 *21-1003 *24-1121 25-1158
P.REL	= ****** GX	24-1120
REP7	003420 R	7-145 7-146 7-147 7-148 7-149 7-150 7-151 7-152 7-154
		7-158 7-159 7-160 7-161 7-167 7-175 7-177 7-178 #27-1244
		28-1265 29-1310 30-1339

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 4

```
                                ;+
                                ; $TIOUT - PRINT MESSAGE ON TI:
                                ; $FNOUT - PRINT NAME OF LAST FILE OPENED ON TI:
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE ADDRESS
                                ;       R1=MESSAGE LENGTH
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000270                          $FNOUT::
000270  012700  000233'                 MOV     #FNERR,R0       ; GET MESSAGE ADDRESS
000274  016701  000030'                 MOV     FNPTR,R1        ; CALCULATE LENGTH
000300  160001                          SUB     R0,R1           ;       ...
000302                          $TIOUT::
000302  010067  000014'                 MOV     R0,TIOUT+Q.IOPL ; SET DPB BUFFER ADDRESS
000306  010167  000016'                 MOV     R1,TIOUT+Q.IOPL+2 ; AND BUFFER LENGTH
000312                                  DIR$    #TIOUT          ; ISSUE TI: OUTPUT QIO
000320                                  RETURN
```

```
NTLSYM - OFFSET SYMBOL DEFINITI MACRO V05.03b Saturday 29-Jun-85 00:34  Page 5-4
Symbol table

AB     = 000000 G        L$$DRV= 000000          ZF.DDM= 000001 G        .APR    000040 G        .MXTAB  000023 G
A$$CHK= 000000           L$$P11= 000001          ZF.DIA= 004000 G        .BLKB   000003 G        .NODE   000024 G
A$$CPS= 000000           L$$11R= 000000          ZF.DLC= 000002 G        .BLKW   000004 G        .PECHA  000045 G
A$$PRI= 000000           M$$CRB= 000124          ZF.DVP= 100000 G        .BYTE   000005 G        .PRI    000025 G
A$$TRP= 000000           M$$CRX= 000000          ZF.INI= 040000 G        .CORE   000006 G        .PRIOV  000026 G
C$$CKP= 000000           M$$FCS= 000000          ZF.KMX= 000020 G        .CSR    000007 G        .SCOM   000027 G
C$$ORE= 000400           M$$MGE= 000000          ZF.LLC= 000004 G        .CTIM   000010 G        .SECSR  000030 G
C$$RSH= 177564           M$$NET= 000000          ZF.LMC= 000100 G        .CXALL  000062 G        .SLNB   000031 G
D$$BUG= 177514           M$$OVR= 000000          ZF.MAN= 020000 G        .CXCSR  000004 G        .SLNW   000032 G
D$$ISK= 000000           N$$ACC= 000001          ZF.MFL= 000010 G        .CXLBR  000060 G        .STNB   000033 G
D$$L11= 000001           N$$BUF= 000001          ZF.MTM= 000400 G        .CXLB1  000030 G        .STNW   000034 G
D$$YNC= 000000           N$$LDV= 000001          ZF.MUX= 000040 G        .CXLB2  000044 G        .TIME   000043 G
D$$YNM= 000000           N$$MCP= 000001          ZF.PSE= 002000 G        .CXLEN  000064 G        .UNB    000035 G
E$$XPR= 000000           N$$MLL= 000001          ZF.SLI= 010000 G        .CXPCB  000006 G        .UNW    000036 G
F$$LVL= 000001           N$$MOV= 000010          ZF.TIM= 000200 G        .CXSYM  000010 G        .WORD   000037 G
G$$TPP= 000000           N$$NCT= 000001          ZF.X3P= 000000 G        .CXUNL  000024 G        .X2CH1  000061 G
G$$TSS= 000000           N$$PEM= 000001          ZS.ASN= 100000 G        .CXVEC  000026 G        .X2CH2  000062 G
G$$TTK= 000000           P$$P45= 000000          ZS.BSY= 140000 G        .CXVSM  000016 G        .X2CH3  000063 G
G$$WRD= 000000           P$$WRD= 000000          Z.AVL   000014 G        .DPRB   000041 G        .X2CH4  000064 G
I$$RAR= 000000           Q$$OPT= 000010          Z.DAT   000016 G        .DPRW   000042 G        .X2CH5  000065 G
I$$RDN= 000000           R$$DER= 000000          Z.DSP   000000 G        .DVCHA  000011 G        .X3C01  000046 G
K$$CNT= 177546           R$$K11= 000001          Z.FLG   000010 G        .EOF    000012 G        .X3C02  000047 G
K$$CSR= 177546           R$$SND= 000000          Z.LEN = 000016 G        .FILE   000013 G        .X3C03  000050 G
K$$LDC= 000000           R$$11M= 000000          Z.LLN   000006 G        .INT    000014 G        .X3C04  000051 G
K$$TPS= 000074           S$$WRG= 000000          Z.MAP   000020 G        .LFILE  000017 G        .X3C05  000052 G
LB.ADR  000006 G         S$$YSZ= 007600          Z.NAM   000004 G        .LIBR   000015 G        .X3C06  000053 G
LB.APR= 000013 G         T$$KMG= 000000          Z.PCB   000012 G        .LINKS  000016 G        .X3C07  000054 G
LB.SIZ  000010 G         T$$MIN= 000000          Z.SCH   000007 G        .LSTHD  000020 G        .X3C08  000055 G
LB.USE  000012 G         V$$CTR= 001000          $KPBLN= 000004 G        .LTAB   000021 G        .X3C09  000056 G
LD$LP = 000000           X$$DBT= 000000          .ADDRB  000001 G        .MPTAB  000022 G        .X3C10  000057 G
L$$ASG= 000000           ZF.COU= 001000 G        .ADDRW  000002 G        .MXPTB  000044 G        .X3C11  000060 G


. ABS.  000070    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10085 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.27
SY:NTLSYM.V2,[132,134]NTLSYM/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLSYM
```



```
NTLSYM     CREATED BY  MACRO  ON 29-JUN-85 AT 00:34     PAGE 1
```

```
                                ;+
                                ; $ELINE - ECHO LAST RECORD FROM TEMPLATE FILE
                                ; $EFNLL - ECHO OPEN FAILURE FILE NAME AND LAST RECORD FROM TEMPLATE FILE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R0,R1=DESTROYED
                                ;-
000326                          $EFNLL::
000326                                  CALL    $FNOUT          ; ECHO FILE NAME
000332                          $ELINE::
000332  012700  000146'                 MOV     #TMPBF,R0       ; GET BUFFER ADDRESS
000336  016701  000352'                 MOV     TMPSZ,R1        ; AND LENGTH
000342                                  CALLR   $TIOUT          ; PRINT IT ON TI:
```

Table of contents

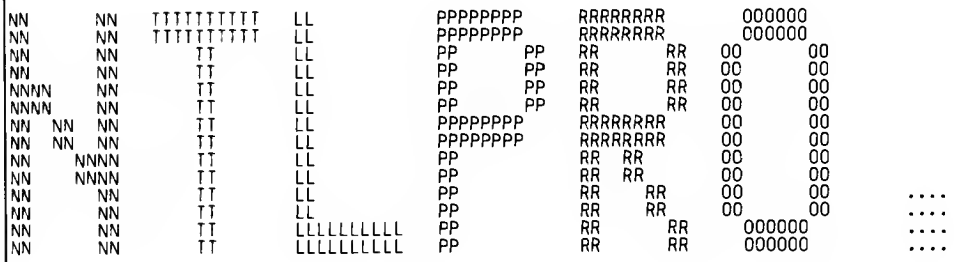

```
     5-   53    MACRO DEFINITIONS
     6-  105    LOCAL SYMBOL DEFINITIONS
     7-  127    LOCAL DATA
     8-  206    $XPRO - CLEAN UP PROCESS
     9-  324    $DSINI  INIT/TERM PROCESS VIA DIRECT CALL
    10-  363    DEAECL - DEALLOCATE ECL NODE COUNTERS
    11-  390    $XLIN - CLEAN UP LINE
    12-  468    $CPRO - CLEAR PROCESS
    13-  503    DQCLK - DEQUEUE CLOCK BLOCK
    14-  548    $CLIN - CLEAR LINE
    15-  666    VECT - RESET INTERRUPT VECTOR(S)
    16-  891    ICBSC - DETERMINE IF VECTOR POINTS AT ICB
    17-  935    ICBLEN - CHECK ICB FOR LENGTH
    18- 1013    ZDDS - CLEAR DDM LINE TABLE ADDRES WORD(S) AND KRB ADDRESS
    19- 1054    $CCEX - CLEAR CEX
    20- 1350    XDCB - UNLOAD DEVICES
    21- 1381    XUMR - RELEASE UMR'S
    23- 1442    XOBJTB - DE-ALLLOCATE OBJECT TABLE
    24- 1459    XALITB - DE-ALLOCATE ALIAS TABLE
    25- 1498    XREMTB - DE-ALLOCATE REMOTE TABLE
    26- 1523    XFLTTB - DE-ALLOCATE FILTER CONTROL BLOCKS
    27- 1556    FLT16 - FIND A 16 BIT ALLOCATION LINE TABLE
    28- 1651    XLT16 - DE-ALLOCATE A 16 BIT ALLOCATION LINE TABLE
    29- 1783    .DEA16 - END OF PARTITION DE-ALLOCATION ROUTINE
    30- 1894    $DEAC2 - DEALLOCATE POOL SPACE TO AREA OUTSIDE TASK SPACE
    31- 1952    DEALST - DEALLOCATE LIST OF DATA BLOCKS
    32- 1988    XSERTB DEALLOCATE SERVICE BLOCKS
```

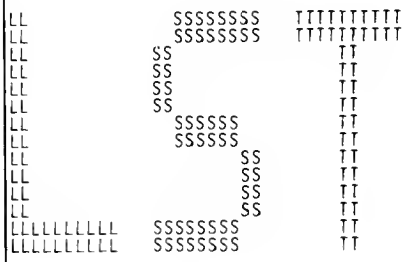
```
    112 000000                          .PSECT
```

```
                        .SBTTL  KMC DATA BASE DESCRIPTION


;
; THE KMC DATA BASE (KMC LOADED LIST) IS A LIST OF ALL KMC'S WHICH
; HAVE HAD THEIR MICRO-CODE LOADED, OR WHICH ARE TO HAVE THEIR
; MICRO-CODE LOADED ON POWER-FAIL OR SYSTEM STARTUP.
;
; THIS LIST CONSISTS OF A SERIES OF LIST ENTRIES, COLLECTED INTO
; 32. WORD BLOCKS.  THE FIRST BLOCK IS POINTED TO BY THE WORD AT
; THE OFFSET ".CXKMC" INTO THE NTL CONTROL DATA BLOCK (POINTED
; TO BY "$NTLPT").  THE FIRST WORD OF EACH BLOCK IS A POINTER
; TO THE NEXT BLOCK (IF ANY).  THERE ARE 7 ENTRIES PER BLOCK,
; AND 4 WORDS PER ENTRY.  THIS LEAVES 3 UNUSED WORDS AT THE
; END OF EACH BLOCK.
;
; THE FORMAT FOR EACH ENTRY IS AS FOLLOWS:
;
;       .RAD50  /NAM/   - THE PDV/DEVICE NAME FOR THE PARTICULAR KMC
;       .WORD   CSR     - THE CSR ADDRESS FOR THE PARTICULAR KMC
;       .BYTE   NUM     - THE CONTROLLER NUMBER FOR THE KMC
;       .BYTE   LEN     - LENGTH OF LOCAL MICRO-CODE COPY IN 32. WORD BLOCKS
;       .WORD   PNT     - ADDRESS (BIAS) OF LOCAL MICRO-CODE COPY
;
; THE LAST TWO PIECES OF DATA IN THE ENTRY (LEN AND PNT) ARE APPLICABLE
; ONLY ON AN RSX11-S SYSTEM WHEN FIRST INITIALIZING THE SYSTEM.  AT ALL
; OTHER TIMES, THEY SHOULD BE ZERO.
;
```

```
                                ;
                                ; CHECK FILE STRUCTURE FOR ERRORS, $CLOPE HAS READ THE ATTRIBUTES AND
                                ; SET "$LBN" FOR $RLBL.  $RLBL SETS R0, "$LLEN", AND "$LBN" FOR THE
                                ; FIRST $READ OF THE TEXT BLOCKS.
                                ;
000654  052010                        BIS     (R0)+,(R0)              ;B-R0 -> INFO, IS FILE CONTIGUOUS?
000656  001513                        BEQ     34$                     ;B-BR IF NO, ERROR
000660  012700  000000'                MOV     #KMCBUF,R0              ;B-SET BUFFER ADDR TO 1000 BYTE BUFFER
000664                                CALL    $RLBL                   ;B-READ LABEL BLOCK, SET R0->BUFFER,
                                                                      ;B-  $LLEN=1000, $LBN->FIRST TEXT BLOCK
000670  103512                        BCS     36$                     ;B-BR IF CAN'T READ LABEL BLOCK
                                .IF     DF,R$$11M
000672  032760  000000G 000000G       BIT     #TS$NHD,L$BFLG(R0)      ;B-DOES FILE HAVE A HEADER?
                                .IFF
                                BIT     #LF$NHD,L$LFLG(R0)      ;B-DOES FILE HAVE A HEADER?
                                .ENDC
000700  001515                        BEQ     38$                     ;B-BR IF YES, ERROR
000702  016003  000000G               MOV     L$BLDZ(R0),R3           ;B-NUMBER OF 32. WORD ROOT BLOCKS
000706  026003  000000G               CMP     L$BMXZ(R0),R3           ;B-IS FILE OVERLAID?
000712  001114                        BNE     40$                     ;B-BR IF YES, ERROR
000714  062703  000007                ADD     #7,R3                   ;B-MUST ROUND UP TO NEXT BLOCK
000720  006203                        ASR     R3                      ;B-CONVERT 32. WORD BLOCKS
000722  006203                        ASR     R3                      ;B-  TO NUMBER OF 256.
000724  006203                        ASR     R3                      ;B-  WORD BLOCKS
                                ;
                                ; INITIALIZE KMC FOR LOAD
                                ;
000726  010401                        MOV     R4,R1                   ;B-COPY KMC CSR ADDRESS
000730  005021                        CLR     (R1)+                   ;B-CLEAR ALL KMC REGISTERS
000732  005021                        CLR     (R1)+
000734  005021                        CLR     (R1)+
000736  005011                        CLR     (R1)
000740  005001                        CLR     R1                      ;B-INITIAL KMC RAM ADDRESS
```

```
**FILE**ID**NTLPRO
```


NN NN TTTTTTTTTT LL PPPPPPPP RRRRRRRR 000000
NN NN TTTTTTTTTT LL PPPPPPPP RRRRRRRR 000000
NN NN TT LL PP PP RR RR 00 00
NN NN TT LL PP PP RR RR 00 00
NNNN NN TT LL PP PP RR RR 00 00
NNNN NN TT LL PP PP RR RR 00 00
NN NN NN TT LL PPPPPPPP RRRRRRRR 00 00
NN NN NN TT LL PPPPPPPP RRRRRRRR 00 00
NN NNNN TT LL PP RR RR 00 00
NN NNNN TT LL PP RR RR 00 00
NN NN TT LL PP RR RR 00 00 ....
NN NN TT LL PP RR RR 00 00 ....
NN NN TT LLLLLLLLLL PP RR RR 000000 ....
NN NN TT LLLLLLLLLL PP RR RR 000000 ....



NTLPRO - "SET PROCESS" ROUTINES MACRO V05.03 Saturday 29-Jun-85 00:31
Table of contents

```
                                        MOV     #VECLEN,R0      ; NOW TAKE THE ASCII NAME
                                        MOV     PC,R1           ; AND CONVERT TO RAD50
                                        CALL    $CAT5           ; STORING THE RAD50 NAME IN THE GLOBAL
                                        MOV     R1,VLSYM        ; SYMBOL LOOKUP CONTROL BLOCK
                                        MOV     PC,R1           ;       ...
                                        CALL    $CAT5           ;       ...
                                        MOV     R1,VLSYM+2      ;       ...
                                        .ENDC

000562  116767  000000G 000013'       MOVB    $$DEV,DSPTAB+1  ; FORM DCB DISPATCH TABLE ADDRESS NAME
000570  116767  000001G 000014'       MOVB    $$DEV+1,DSPTAB+2;       ...
000576  012700  000012'               MOV     #DSPTAB,R0      ; CONVERT TO RAD50
000602  010701                        MOV     PC,R1           ;       ...
000604                                CALL    $CAT5           ;       ...
000610  010167  000062'               MOV     R1,DTSYM        ; DEFAULT SECOND HALF OF NAME TO "TBL"
000614  005000                        CLR     R0              ; SETUP ASCIZ FILE SPEC
000616  005001                        CLR     R1              ; FOR [131,5XJXXX.STB

000620  012702  000006'               MOV     #STBEXT,R2      ;       ...
000624                                CALL    $AZFS           ;       ...
000630  012700  000020'               MOV     #PBLK,R0        ; GET LOOKUP CONTROL BLOCK ADDRESS
000634                                CALL    $SYMBL          ; LOOK FOR GLOBAL SYMBOL
000640  103446                        BCS     STBERR          ; IF CS, FILE ERROR
                                                              ; NOTE - R0 IS NOW 0

                                        .IF DF  R$$MPL

                                        MOV     VECVAL,$VCTB    ; COPY VECTOR TABLE ADDRESS
                                        MOV     VLNVAL,$VCLN    ; COPY VECTOR TABLE LENGTH
                                        BIC     #160000,$VCTB   ; BIAS VECTOR TABLE THRU APR6
                                        BIS     #140000,$VCTB   ; ...
                                        .ENDC

000642  016767  000066' 000000G       MOV     DTVAL,$$TBL     ; COPY DCB DISPATCH TABLE ADDRESS
000650  005767  000022'               TST     PBLK+2          ; IS PROCESS DISPATCH TABLE DEFINED?
000654  001432                        BEQ     101$            ; IF EQ, NO

000656                                SWSTK$  70$             ;* ENTER KERNEL MODE
000662  017700  000000G               MOV     @NTLPT,R0       ;* GET DATA BLOCK BIAS
000666                                CALL    $MAPX           ;* MAP TO DATA BLOCK
000672  005000                        CLR     R0              ;* CLEAR ERROR INDICATOR

000674  062702  000000G               ADD     #.CXSYM,R2      ;* OFFSET TO SYMBOL AREA

                                        .IF DF  R$$MPL

                                        TST     VECVAL          ;* PROCESS VECTORED ?
                                        BEQ     38$             ;* IF NO - BRANCH
                                        ADD     #<.CXVSM-.CXSYM>,R2 ;* IF YES - USE VECTORED VALIDATION SYMBOLS
                                        .ENDC

000700  012701  000126'       38$:    MOV     #XBLK1,R1       ;* GET FIRST SYMBOL BLOCK ADDRESS
000704  026122  000010         40$:    CMP     10(R1),(R2)+    ;* DOES SYMBOL MATCH COMM EXEC VALUE?
000710  001406                        BEQ     50$             ;* IF EQ, YES
000712  005761  000002                TST     2(R1)           ;* IS SYMBOL DEFINED?

000716  001403                        BEQ     50$             ;* IF EQ, NO
000720  005266  000004                INC     4(SP)           ;* INDICATE SYMBOL MISMATCH
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ADJDF$	#5-59	5-61								
ASL$	#5-58	7-297	7-301							
ASR$	#5-58	12-641	13-693							
BIAS	#5-58	12-653	13-707	13-713	13-723					
BIAS5	#5-87									
CACDF$	#5-59	5-67								
CALL	7-276	7-277	7-283	7-284	7-304	7-306	7-312	10-480	10-483	10-523
	10-529	10-531	10-546	10-548	11-599	11-600	11-602	11-604	12-629	12-642
	13-683	13-689	13-695	13-702	13-703					
CALLR	7-323	10-584								
CRASH	7-320									
CTRDF$	#5-58	5-62								
DHBDF$	#5-59	5-66								
ECDDB$	#5-58	5-63								
EMSG$	#5-58									
EMSG$R	#5-58	7-321	10-581	10-582	12-664	13-727				
MAP	#5-92									
RESMAP	#5-83									
RESRG	#5-59									
RETC	#5-58									
RETURN	7-310	7-314	8-372	9-443	10-572	10-576	11-606	12-662	13-725	
SAVMAP	#5-79									
SAVRG	#5-59									
SLTDF$	#5-58	5-65								
SOB	10-476	13-717								
SWSTK$	7-304	10-546	13-702							
XPDDB$	#5-58	5-64								

FILEID**NTLROO

```
                                        .SBTTL  $SPRO - SET PROCESS

                                ;+
                                ;
                                ; $SPRO - "SET PROCESS"
                                ;
                                ; INPUTS:
                                ;       $PDVA=PDV ADDRESS
                                ;       $NAME=RAD50 PDV NAME
                                ;       $$INC=INCREMENT VALUE (OPTIONAL)
                                ;       $$PAR=PARTITION NAME (OPTIONAL)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;
                                ;-
000070                          $SPRO:
000070  016700  000062'                MOV     $PDVA,R0        ; GET PDV ADDRESS
000074  016067  000000G 000070'        MOV     Z.PCB(R0),$PCB  ; IS PROCESS CORE-RESIDENT?
000102  001006                        BNE     10$             ; IF NE, YES
000104  004767  000000G               CALL    $XFER           ; TRANSFER PROCESS FROM DISK TO CORE
000110  103476                        BCS     70$             ; IF CS, ERROR (NOTHING TO CLEAN UP)

                                        .IF DF  R$$MPL
                                        CALL    VECSET          ; INIT PROCESS VECTOR TABLE
                                        .ENDC

000112  052767  010000  001232'        BIS     #FL.PX1,$FLAGS  ; INDICATE I/O TRANSFER OCCURRED

000120  016701  000062'        10$:    MOV     $PDVA,R1        ; GET PDV ADDRESS OF PROCESS
000124  032761  000000G 000000G        BIT     #ZF.PSE,Z.FLG(R1) ; IS THIS DLM?
000132  001023                        BNE     40$             ; BR IF YES - DON'T EXPAND LINE TABLE
000134  032767  000004  001232'        BIT     #FL.LLC,$FLAGS  ; ARE WE LOADING AN LLC?
000142  001403                        BEQ     30$             ; IF EQ, NO
000144  004767  000000G               CALL    $TMP            ; SETUP LLC LINE TABLE
000150  103457                        BCS     101$            ; IF CS, FAILURE

000152  016701  000062'        30$:    MOV     $PDVA,R1        ; GET PDV ADDRESS

000156  026127  000000G 114224  35$:    CMP     Z.NAM(R1),#^RXPT ; ARE WE LOADING XPT?
000164  001006                        BNE     40$             ; BR IF NO
000166  004767  000000G               CALL    $INICO          ; ALLOC XPT COUNTERS,ROL AND NI CACHE
000172  103446                        BCS     101$            ; IF ERROR - BRANCH

000174  004767  000000G               CALL    $ALMHC          ; ALLOCATE MIN HOPS/MIN COST
000200  103443                        BCS     101$            ; BR IF FAILURE

000202  026127  000000G 017704  40$:    CMP     Z.NAM(R1),#^RECL ; ARE WE LOADING ECL?
000210  001003                        BNE     50$             ; BR IF NO
000212  004767  000000G               CALL    $ALECL          ; ALLOCATE ECL NODE COUNTERS
000216  103434                        BCS     101$            ; BR IF FAILURE

000220  005767  000100'        50$:    TST     $$DEV           ; DEVICE NAME INCLUDED?
000224  001403                        BEQ     60$             ; IF EQ, NO
000226  004767  000000G               CALL    $NLDEV          ; SETUP DEVICE DATA STRUCTURES
000232  103426                        BCS     101$            ; IF CS, FAILURE
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES
KSAR7		000012	RG	#5-89
LDBAF		000126	RG	#5-153
LDPWF		000052	RG	#5-121
LF.ACT	=	100000		#4-61
LF.BRO	=	000400		#4-61
LF.BWT	=	000007		#4-61
LF.ENA	=	002000		#4-61
LF.LPB	=	001000		#4-61
LF.MDC	=	000100		#4-61
LF.MFL	=	004000		#4-61
LF.MTP	=	000020		#4-61
LF.PAC	=	000200		#4-61
LF.RDY	=	040000		#4-61 14-826
LF.REA	=	010000		#4-61
LF.SER	=	000040		#4-61
LF.TIM	=	000010		#4-61
LF.UNL	=	020000		#4-61
LF.X2P	=	000000		#4-61
LGDDB		000206	RG	#6-183
LGPDV		000204	RG	#6-182
LGSTT		000154	RG	#6-170
LLCAL		000200	RG	#6-180
LLCTA		000136	RG	#5-157
LN.CLO	=	000000		#4-61
LN.DUM	=	000005		#4-61
LN.LOA	=	000004		#4-61
LN.LOO	=	000003		#4-61
LN.OAU	=	000003		#4-61
LN.OFF	=	000001		#4-61
LN.ON	=	000000		#4-61
LN.OOP	=	000004		#4-61
LN.OPE	=	000001		#4-61
LN.REF	=	000002		#4-61
LN.SER	=	000002		#4-61
LN.STA	=	000017		#4-61
LN.SUB	=	000360		#4-61
LN.TRI	=	000006		#4-61
LOOP		000004	R	#7-472 7-487
LTMFC		000120	RG	#5-150
L.COST		000015		#4-61
L.CTL		000012		#4-61
L.CVA		177776		#4-61
L.DDM		000002		#4-61
L.DDS		000004		#4-61
L.DLC		000003		#4-61
L.DLM		000006		#4-61
L.DLS		000010		#4-61
L.FLG		000000		#4-61 14-826
L.KRBA		000016		#4-61
L.LEN	=	000022		#4-61
L.MPF		000022		#4-61
L.NMST		000020		#4-61

```
                                        .SBTTL  LOCAL DATA

                                ;
                                ; LOCAL DATA
                                ;

000000                                  .PSECT  DATA,D

                                        .NLIST  BEX

                                ;
                                ; LOOKUP TABLE USING FUNCTION CODE AS INDEX
                                ;
                                FTAB:   .WORD   .SCEX           ; 1 - SET CEX (NTINIT)
                                        .WORD   .SPRO           ; 2 - SET PROCESS
                                        .WORD   .SLIN           ; 3 - SET LINE
                                        .WORD   .CCEX           ; 4 - CLEAR CEX
                                        .WORD   .CPRO           ; 5 - CLEAR PROCESS
                                        .WORD   .CLIN           ; 6 - CLEAR LINE
                                        .WORD   .KMCPF          ; 7 - RELOAD KMC MICROCODE AFTER POWERFAIL
                                        .WORD   .SCIR           ; 10 - SET CIRCUIT
                                FMAX=.-FTAB/2

                                ;
                                ; SUBROUTINE TABLES
                                ;
                                .SCEX:  .WORD   CLRPP           ; CLEAR PROCESS PARAMETERS
                                        .WORD   SCEX,0
                                .SPRO:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   PDVA            ; FIND PDV ADDRESS
                                        .WORD   CHKPP           ; CHECK PROCESS PARAMETERS
                                        .WORD   SPRO,0
                                .SLIN:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   SLTA            ; FIND SLT ADDRESS
                                        .WORD   CHKLP           ; CHECK LINE PARAMETERS
                                        .WORD   CLRPP           ; CLEAR PROCESS PARAMETERS
                                        .WORD   SLIN,0
                                .CCEX:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   CCEX,0
                                .CPRO:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   PDVA            ; FIND PDV ADDRESS
                                        .WORD   CPRO,0
                                .CLIN:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   SLTA            ; FIND SLT ADDRESS
                                        .WORD   CLIN,0
                                .KMCPF: .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   KMCPF,0
                                .SCIR:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
                                        .WORD   SCIR,0          ;


                                NULL:   .WORD   0,0,0,0         ; NULL PARAMETER BLOCK
                                PDVX:   .BLKW   1               ; DDM PDV INDEX (SET LINE)
                                UNITS:  .BLKW   1               ; NUMBER OF UNITS ON CONTROLLER
                                SPSAV:  .BLKW   1               ; STACK POINTER SAVE
                                PDVNUM: .BLKW   1               ; NUMBER OF PDV ENTRIES
```



```
000000  000020'                 FTAB:   .WORD   .SCEX           ; 1 - SET CEX (NTINIT)
000002  000026'                         .WORD   .SPRO           ; 2 - SET PROCESS
000004  000040'                         .WORD   .SLIN           ; 3 - SET LINE
000006  000054'                         .WORD   .CCEX           ; 4 - CLEAR CEX
000010  000062'                         .WORD   .CPRO           ; 5 - CLEAR PROCESS
000012  000072'                         .WORD   .CLIN           ; 6 - CLEAR LINE
000014  000102'                         .WORD   .KMCPF          ; 7 - RELOAD KMC MICROCODE AFTER POWERFAIL
000016  000110'                         .WORD   .SCIR           ; 10 - SET CIRCUIT
        000010                  FMAX=.-FTAB/2

                                ;
                                ; SUBROUTINE TABLES
                                ;
000020  001244'                 .SCEX:  .WORD   CLRPP           ; CLEAR PROCESS PARAMETERS
000022  000070'  000000                 .WORD   SCEX,0
000026  000164'                 .SPRO:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000030  000204'                         .WORD   PDVA            ; FIND PDV ADDRESS
000032  000262'                         .WORD   CHKPP           ; CHECK PROCESS PARAMETERS
000034  000404'  000000                 .WORD   SPRO,0
000040  000164'                 .SLIN:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000042  000464'                         .WORD   SLTA            ; FIND SLT ADDRESS
000044  000730'                         .WORD   CHKLP           ; CHECK LINE PARAMETERS
000046  001244'                         .WORD   CLRPP           ; CLEAR PROCESS PARAMETERS
000050  001256'  000000                 .WORD   SLIN,0
000054  000164'                 .CCEX:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000056  002156'  000000                 .WORD   CCEX,0
000062  000164'                 .CPRO:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000064  000204'                         .WORD   PDVA            ; FIND PDV ADDRESS
000066  001306'  000000                 .WORD   CPRO,0
000072  000164'                 .CLIN:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000074  000464'                         .WORD   SLTA            ; FIND SLT ADDRESS
000076  002054'  000000                 .WORD   CLIN,0
000102  000164'                 .KMCPF: .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000104  002120'  000000                 .WORD   KMCPF,0
000110  000164'                 .SCIR:  .WORD   CEXLD           ; MAKE SURE COMM EXEC IS LOADED
000112  002656'  000000                 .WORD   SCIR,0          ;


000116  000000  000000  000000  NULL:   .WORD   0,0,0,0         ; NULL PARAMETER BLOCK
000126                          PDVX:   .BLKW   1               ; DDM PDV INDEX (SET LINE)
000130                          UNITS:  .BLKW   1               ; NUMBER OF UNITS ON CONTROLLER
000132                          SPSAV:  .BLKW   1               ; STACK POINTER SAVE
000134                          PDVNUM: .BLKW   1               ; NUMBER OF PDV ENTRIES
```

```
548 001060  001060                          BNE     121$            ; IF NE, NO
549 001062  026700  000000G                 CMP     $MXVCT,R0       ; IS IT WITHIN THE SYSTEM VECTOR SPACE ?
550 001066  101463                          BLOS    141$            ;  NO
551 001070  042767  000000G 000000G        BIC     #MS.VCT,$MISS   ; INDICATE VECTOR INCLUDED
552 001076  010067  000000G         50$:    MOV     R0,$$VECT       ; SET INTERRUPT VECTOR
553 001102  012300                          MOV     (R3)+,R0        ; GET SECONDARY CSR VALUE
554 001104  001414                          BEQ     70$             ; IF ZERO, FIELD MISSING
555 001106  022700  160000                  CMP     #160000,R0      ; IS IT AN I/O PAGE ADDRESS?
556 001112  101035                          BHI     101$            ; IF HI, NO
557 001114  032700  000003                  BIT     #3,R0           ; IS IT A MULTIPLE OF 4?
558 001120  001032                          BNE     101$            ; IF NE, NO
559
560                                         .IF DF  R$$MPL
561
562                                         TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
563                                         BNE     60$             ; IF NE, YES
564
565                                         .ENDC
566
567 001122                                  CALL    $NLCSR          ; IS THE DEVICE ONLINE ?
568 001126  103440                          BCS     131$            ; IF CS, NO
569 001130  042767  000000G 000000G 60$:    BIC     #MS.SEC,$MISS   ; NO LONGER MISSING
570 001136  010067  000000G         70$:    MOV     R0,..SCSR       ; SAVE SECONDARY CSR
571 001142  012767  000001  000000G         MOV     #1,$$MTP        ; DEFAULT TO ONE STATION
572 001150  112767  000001  000000G         MOVB    #1,$$SNUM       ; AND STATION NUMBER 1 (POINT-TO-POINT)
573 001156  012700  000060                  MOV     #48.,R0         ; MAXIMUM NUMBER OF STATIONS
574 001162  012701  000000G                 MOV     #$$HTIM,R1      ; POINT TO HELLO TIMER
575 001166  012702  000000G                 MOV     #$$LTIM,R2      ; POINT TO LISTEN TIMER
576 001172  005021                  80$:    CLR     (R1)+           ; INITIALIZE HELLO TIMER
577 001174  005022                          CLR     (R2)+           ; INITIALIZE LISTEN TIMER
578 001176  005300                          DEC     R0              ; MORE TO INITIALIZE?
579 001200  003374                          BGT     80$             ; BR IF YES
580 001202  000241                          CLC                     ; INDICATE SUCCESS
581 001204                                  RETURN
582
583                                 ;
584                                 ; ERROR CONDITIONS
585                                 ;
586 001206                  101$:   EMSG$R  94              ; ILLEGAL CSR VALUE
587 001214                  111$:   EMSG$R  93              ; ILLEGAL PRIORITY VALUE
588 001222                  121$:   EMSG$R  92              ; ILLEGAL VECTOR VALUE
589 001230                  131$:   EMSG$R  Z1              ; DEVICE OFFLINE
590 001236                  141$:   EMSG$R  ZP              ; VECTOR NOT IN SYSTEM
```

```
                                        .SBTTL  $FPDV - FIND PDV
                                ;+
                                ;
                                ; $FPDV - TRY TO FIND PDV
                                ;
                                ; INPUTS:
                                ;       R0=RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=PDV ADDRESS
                                ;       R1,R2=DESTROYED
                                ;
                                ;-
003050                          $FPDV::
003050  010046                          MOV     R0,-(SP)        ; SAVE NAME TO BE MATCHED
003052  017701  000000G                 MOV     @PDVTA,R1       ; GET PDV TABLE ADDRESS
                                ;
003056  017702  000000G                 MOV     @PDVNM,R2       ; AND NUMBER OF PDV'S
003062  001407                          BEQ     30$             ; IF ZERO, NO PDV'S!
003064  012100                  10$:    MOV     (R1)+,R0        ; GET PDV ADDRESS
003066  001403                          BEQ     20$             ; IF ZERO, SKIP THIS ONE
                                ;
003070  026016  000000G                 CMP     Z.NAM(R0),(SP)  ; DOES THE NAME MATCH?
003074  001403                          BEQ     40$             ; IF EQ, YES (C-BIT CLEAR)
003076                          20$:    SOB     R2,10$          ; LOOP $PDVNM TIMES
003102  000261                  30$:    SEC                     ; INDICATE FAILURE
003104  005226                  40$:    INC     (SP)+           ; PURGE STACK
003106  006016                          ROR     (SP)            ; SAVE C-BIT
003110  005741                          TST     -(R1)           ; CALCULATE PDV INDEX
003112  167701  000000G                 SUB     @PDVTA,R1       ;       ...
                                ;
003116  016002  000000G                 MOV     Z.PCB(R0),R2    ; GET PCB ADDRESS FOR THIS PROCESS
003122  016267  000000G 000000G         MOV     P.REL(R2),$BIAS ; SAVE BIAS FOR LATER
003130  010167  000126'                 MOV     R1,PDVX         ; SAVE PDV INDEX VALUE
003134  006316                          ASL     (SP)            ; RESTORE C-BIT
003136                                  RETURN
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
REP7A	003422 R	7-168	#28-1264						
REP7B	003564 R	7-176	#30-1338						
R$$EIS	= ******	15-543							
R$$MPL	= ******	7-163	9-264	15-527	15-560	18-663			
R$$11D	= ******	15-543							
R$$11M	= 000000	7-153	7-171	20-830	21-948				
SCEX	000070 R	6-91	#9-260						
SCIR	002656 R	6-112	#22-1026						
SF.ACT	= 000200	#5-62	18-711						
SF.ENA	= 000100	#5-62							
SF.LPB	= 000004	#5-62							
SF.MFL	= 000040	#5-62							
SF.PAC	= 000020	#5-62							
SF.REA	= 000010	#5-62							
SF.SER	= 000001	#5-62							
SF.SVC	= 000002	#5-62							
SF.UNL	= 000040	#5-62							
SLIN	001256 R	6-100	#17-625						
SLTA	000464 R	6-97	6-107	#14-440					
SLTMA	= ****** GX	18-687	18-764	21-928	21-988	25-1149			
SLTNM	= ****** GX	18-766	21-990	25-1156					
SPRO	000404 R	6-95	#13-405						
SPSAV	000132 R	#6-118	*8-219	25-1186					
S$$BAS	= ******	7-145	7-145	7-146	7-146	7-147	7-147	7-148	7-148	7-149
		7-149	7-150	7-150	7-151	7-151	7-152	7-152	7-154	7-154
		7-158	7-158	7-159	7-159	7-160	7-160	7-161	7-161	7-167
		7-167	7-168	7-168	7-169	7-169	7-172	7-172	7-175	7-175
		7-176	7-176	7-177	7-177	7-178	7-178			
S.COST	000001	#5-62								
S.FLG	000000	#5-62								
S.LEN	000004	#5-62								
S.NMST	000002	#5-62								
S.OWNR	000003	#5-62								
UNITS	000130 R	#6-117	*25-1152	25-1162	25-1164	*25-1167	25-1178			
US.BSY	= ****** GX	18-737	21-961							
US.MNT	= ****** GX	18-741	21-965							
U.CWT	= ****** GX	18-735	21-959							
U.STS	= ****** GX	18-737	18-741	21-961	21-965					
ZF.DDM	= ****** GX	14-446								
ZF.DLC	= ****** GX	14-476								
ZF.KMX	= ****** GX	14-457								
ZF.LMC	= ****** GX	14-449								
ZF.MUX	= ****** GX	14-453								
Z.DAT	= ****** GX	13-416								
Z.FLG	= ****** GX	11-336	14-446	14-449	14-453	14-457	14-476	21-914	21-985	22-1036
Z.LLN	= ****** GX	18-677	21-917							
Z.MAP	= ****** GX	18-680	21-921	28-1272						
Z.NAM	= ****** GX	24-1110	28-1286	30-1350						
Z.PCB	= ****** GX	13-412	18-656	18-731	21-911	21-955	24-1119			
$BIAS	= ****** GX	*24-1120								
$BITS	003016 R	13-406	17-626	22-1027	#23-1072					
$CBOMG	= ****** GX	28-1278	30-1362	30-1368						

```
                        ;+
                        ; SLFNP - SETUP LOCAL FILE NAME POINTER
                        ;
                        ; INPUTS:
                        ;       R0=NUMBER TO ADD TO GCL LEVEL INDICATOR (0=CURRENT, 1=PREVIOUS)
                        ;
                        ; OUTPUTS:
                        ;       R0=LOCAL FILE NAME AREA ADDRESS
                        ;-
000322  010046          SLFNP:  MOV     R0,-(SP)        ; SAVE CURRENT/PREVIOUS INDICATOR
000324  116700  000000G        MOVB    $CLEVL,R0       ; GET CURRENT GCL LEVEL
000330  062600                  ADD     (SP)+,R0        ; ADJUST LEVEL BY ORIGINAL R0 VALUE
000332  006300                  ASL     R0              ; 2X
000334  010046                  MOV     R0,-(SP)        ; SAVE 2X VALUE
000336  006300                  ASL     R0              ; 4X
000340  006300                  ASL     R0              ; 8X
000342  062600                  ADD     (SP)+,R0        ; 10X
000344  062700  000113'        ADD     #NAMES-10.,R0   ; OFFSET INTO LOCAL STORAGE AREA
000350                          RETURN
```

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES
AB	= 000000	G	#5-159
I$$AS	= ******		5-64
LB.ADR	000006	G	#5-174
LB.APR	= 000013	G	#5-177
LB.SIZ	000010	G	#5-175
LB.USE	000012	G	#5-176 5-177
R$$MPL	= ******		5-160
R$$11D	= ******		5-64
R$$11M	= 000000		5-64
R$$11S	= ******		5-64
X$$MCB	= ******		5-64 5-64
ZF.COU	= 001000	G	#5-64
ZF.DDM	= 000001	G	#5-64
ZF.DIA	= 004000	G	#5-64
ZF.DLC	= 000002	G	#5-64
ZF.DVP	= 100000	G	#5-64
ZF.INI	= 040000	G	#5-64
ZF.KMX	= 000020	G	#5-64
ZF.LLC	= 000004	G	#5-64
ZF.LMC	= 000100	G	#5-64
ZF.MAN	= 020000	G	#5-64
ZF.MFL	= 000010	G	#5-64
ZF.MTM	= 000400	G	#5-64
ZF.MUX	= 000040	G	#5-64
ZF.PSE	= 002000	G	#5-64
ZF.SLI	= 010000	G	#5-64
ZF.TIM	= 000200	G	#5-64
ZF.X3P	= 000000	G	#5-64
ZS.ASN	= 100000	G	#5-64
ZS.BSY	= 140000	G	#5-64
Z.AVL	000014	G	#5-64
Z.DAT	000016	G	#5-64
Z.DSP	000000	G	#5-64 5-64
Z.FLG	000010	G	#5-64
Z.LEN	= 000016	G	#5-64
Z.LLN	000006	G	#5-64
Z.MAP	000020	G	#5-64
Z.NAM	000004	G	#5-64
Z.PCB	000012	G	#5-64
Z.SCH	000007	G	#5-64
$KRBLN	= 000004	G	#5-164
.ADDRB	000001	G	#5-78
.ADDRW	000002	G	#5-79
.APR	000040	G	#5-109
.BLKB	000003	G	#5-80
.BLKW	000004	G	#5-81
.BYTE	000005	G	#5-82
.CORE	000006	G	#5-83
.CSR	000007	G	#5-84
.CTIM	000010	G	#5-85
.CXALL	000062	G	#5-151
.CXCSR	000004	G	#5-142

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 2

```
                                        ;+
                                        ; $RQ(N) - REQUEST ACCESS TO BINARY BUFFER
                                        ;
                                        ; THESE ROUTINES ARE CALLED TO REQUEST ACCESS TO THE BINARY BUFFER,
                                        ; WHICH IS LOCATED AT THE END OF THE TASK'S SPACE.  THE CALLER SPECIFIES
                                        ; THE NUMBER OF BYTES HE WISHES TO ADD TO THE BUFFER BY USING THE
                                        ; APPROPRIATE ENTRY POINT.
                                        ;
                                        ; INPUTS:
                                        ;       $PTR=CURRENT POINTER INTO BINARY BUFFER
                                        ;       $LEFT=SPACE REMAINING IN BINARY BUFFER
                                        ;       F1.BBO SET IF BUFFER ALREADY OVERFLOWED
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT=SUCCESS/FAILURE
                                        ;       R2=POINTER INTO BINARY BUFFER
                                        ;       $PTR INCREMENTED BY CALLER'S REQUESTED SIZE
                                        ;       $LEFT DECREMENTED BY CALLER'S REQUESTED SIZE
                                        ;       F1.BBO, F1.ERR SET IF THIS REQUEST OVERFLOWS BUFFER
                                        ;-
                                                .ENABL  LSB
000346  012746  000007          $RQ7::  MOV     #7,-(SP)
000352  000416                  $RQ6::  BR      10$
000354  012746  000005          $RQ5::  MOV     #5,-(SP)
000360  000413                  $RQ4::  BR      10$
000362  012746  000004          $RQ4::  MOV     #4,-(SP)
000366  000410                  $RQ3::  BR      10$
000370  012746  000003          $RQ3::  MOV     #3,-(SP)
000374  000405                  $RQ2::  BR      10$
000376  012746  000002          $RQ2::  MOV     #2,-(SP)
000402  000402                  $RQ1::  BR      10$
000404  012746  000001          $RQ1::  MOV     #1,-(SP)
000410  032767  000001  000000' 10$:    BIT     #F1.BBO,$FLAG1  ; HAS BUFFER ALREADY OVERFLOWED?
000416  001011                          BNE     20$             ; IF NE, YES
000420  161667  000010'                 SUB     (SP),$LEFT      ; ENOUGH ROOM FOR THIS REQUEST?
000424  002010                          BGE     30$             ; IF GE, YES
000426                                  EMSG$   13              ; BINARY BUFFER OVERFLOW
000434  052767  000001  000000'        BIS     #F1.BBO,$FLAG1  ; INDICATE OVERFLOW
000442  005426                  20$:    NEG     (SP)+           ; PURGE STACK (SETS C-BIT)
000444  000410                          BR      40$
000446  005767  000000'          30$:    TST     $FLAG1          ; HAVE ANY OTHER ERRORS OCCURRED?
000452  100773                          BMI     20$             ; IF MI, YES
000454  016702  000006'                 MOV     $PTR,R2         ; RETURN CURRENT POINTER VALUE
000460  062667  000006'                 ADD     (SP)+,$PTR      ; ADJUST $PTR UP
000464  105012                          CLRB    (R2)            ; STORE EOF INDICATOR (CLEARS C-BIT)
000466                          40$:    RETURN

                                                .DSABL  LSB
```

```
                        .TITLE  NTLUNL - NTL UNLOAD ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - UNLOAD ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                                ;+
                                ; $NLINI - INITIALIZE NETLDR
                                ;
                                ; INPUTS:
                                ;       NO REGISTER SAVING IS NECESSARY
                                ;
                                ; OUTPUTS:
                                ;       $XBUF2=ADDRESS OF END OF TASK BUFFER
                                ;       $XLEN2=LENGTH OF END OF TASK BUFFER
                                ;-
000000                          $NLINI::
                                        .IF NDF R$$MPL
000000                                  RSXOK$                  ; CHECK FOR RSX COMPATIBILITY
000030  103433                          BCS     101$            ; IF CS, INCOMPATIBLE
                                        .ENDC

000032                                  CALL    $LDUIC          ; DETERMINE DISK AND UIC ASSIGNMENTS
000036  012701  000000G                 MOV     #$CXUIC,R1      ; POINT AT COMM EXEC UIC LOCATION
000042  012702  000000G                 MOV     #$DFUIC,R2      ; POINT AT DEFAULT UIC LOCATION
000046  012703  000000G                 MOV     #$CFUIC,R3      ; POINT AT CONFIG FILE UIC LOC

000052  111021                  10$:    MOVB    (R0),(R1)+      ; COPY THE NEW UIC INTO THE
000054  111022                          MOVB    (R0),(R2)+      ; THREE GLOBAL UIC LOCATIONS
000056  112023                          MOVB    (R0)+,(R3)+     ; UNTIL A ZERO BYTE IS FOUND
000060  001374                          BNE     10$             ;       ...

000062                          20$:    CALL    $XXINI          ; INITIALIZE THE END OF TASK BUFFER
000066  016767  000000G 000000G         MOV     $XXBUF,$XBUF2   ; COPY BUFFER START ADDRESS
000074  016767  000000G 000000G         MOV     $XXLEN,$XLEN2   ; AND BUFFER LENGTH
000102  042767  000000C 000000G         BIC     #^C<FL.ECH!FL.TOP!FL.WRN>,$FLAGS ; SAVE GLOBAL NETLDR STATUS
000110  016767  000000G 000002G         MOV     $FLAGS,$FLAGS+2 ;       ...
000116                                  RETURN

                                ;
                                ;  ERROR CONDITION
                                ;
000120                          101$:   EMSG$   40              ; PRINT ERROR MESSAGE
000126  012767  177777  000000G         MOV     #-1,$FUNC       ; INDICATE FAILURE
000134                          103$:   CALL    $NLGET          ; GET ALL REQUEST BLOCKS
000140                                  CALL    $NLPUT          ; AND RETURN WITH FAILURE CODE
000144  000773                          BR      103$
```

```
                                .SBTTL  LOCAL DATA AND DEFINITIONS

000000                          .PSECT  DATA,D

                        ;
                        ; SET AFFINITY DPB
                        ;
                                .IF DF  R$$MPL
                        SETAF:  STAF$   0
                                .ENDC

                        ;
                        ; DATA NEEDED BY THE POWER-FAIL RELOAD ROUTINE
                        ;
                        ;  KMCNAM         - HOLDS RAD50 NAME OF DEVICE TO LOAD (MAPPED ONLY)
                        ;  KMCCSR         - HOLDS CSR ADDRESS OF DEVICE TO LOAD (MAPPED ONLY)
                        ;  KMCMP1,KMCPN1  - HOLDS MAPPING/POINTER TO CURRENT ENTRY (MAPPED ONLY)
                        ;  KMCNUM         - HOLDS NUMBER OF ENTRIES LEFT IN CURRENT BLOCK (MAPPED ONLY)
                        ;
                        ; DATA NEEDED BY THE KMC ''LOAD LIST'' HOUSEKEEPING ROUTINE
                        ;
                        ;  KMCMP1,KMCPN1  - RETURNS: MAPPING/POINTER TO LINK TO NEW BLOCK
                        ;  KMCNAM         - HOLDS RAD50 NAME OF DEVICE BEING SEARCHED FOR
                        ;  KMCNUM         - HOLDS CONTROLLER NUMBER FOR DEVICE
                        ;  KMCCSR         - HOLDS CSR ADDRESS FOR DEVICE
                        ;
                        ; DATA NEEDED BY RAM ERROR REPORTING ROUTINES
                        ;
                        ;  KMCDEV         - INPUT: ASCIZ PDV/DEVICE NAME (KMCRL1 ONLY)
                        ;  KMCCSR         - INPUT: CSR ADDRESS FOR BAD DEVICE (KMCRL1 ONLY)
                        ;  KMCER1         - INPUT: OCTAL ARGUMENT 1 (KMCRL1 ONLY)
                        ;  KMCER2         - INPUT: OCTAL ARGUMENT 2 (KMCRL2 ONLY)
                        ;  KMCER3         - INPUT: OCTAL ARGUMENT 3 (KMCRL2 ONLY)
                        ;

000000                  KMCNAM: .BLKW   1
000002                  KMCCSR: .BLKW   1
000004                  KMCNUM: .BLKW   1
000006                  KMCDEV: .BLKW   2
                                .IF DF  M$$MGE
000012                  KMCMP1: .BLKW   1
                                .ENDC
000014                  KMCPN1: .BLKW   1
000016                  KMCER1: .BLKW   1
000020                  KMCER2: .BLKW   1
000022                  KMCER3: .BLKW   1

        000001          .IIF    DF,M$$MGE,      KM.ALL= 1       ;M-ALLOCATE ONE 32. WORD BLOCK
                        .IIF    NDF,M$$MGE,     KM.ALL= 100     ;U-ALLOCATE 32. WORDS
```

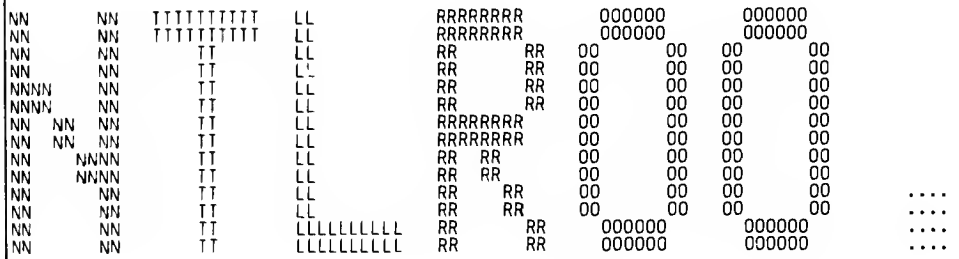
```
                                        ;
                                        ; READ ALL TEXT BLOCKS FROM FILE, MOVE DATA INTO KMC RAM
                                        ;
000742                          10$:    CALL    $READ                   ;B-READ A BLOCK FROM FILE
000746  103502                          BCS     44$                     ;B-BR ON READ ERROR
000750  010046                          MOV     R0,-(SP)                ;B-SAVE BUFFER ADDRESS
000752  012702  000400                  MOV     #400,R2                 ;B-SET WORD COUNT
000756  012714  002000          20$:    MOV     #2000,(R4)              ;B-SELECT CRAM
000762  010164  000004                  MOV     R1,4(R4)                ;B-LOAD ADDRESS ON KMC RAM
000766  011064  000006                  MOV     (R0),6(R4)              ;B-LOAD WORD OF MICRO-CODE
000772  052714  020000                  BIS     #20000,(R4)             ;B-PLACE DATA IN KMC RAM
000776  005014                          CLR     (R4)                    ;B-CLEAR KMC CSR
001000  005064  000004                  CLR     4(R4)                   ;B-CLEAR RAM ADDRESS
001004  005064  000006                  CLR     6(R4)                   ;B-CLEAR DATA
001010  012714  002000                  MOV     #2000,(R4)              ;B-READ CONTENTS OF ADDRESS
001014  010164  000004                  MOV     R1,4(R4)                ;B-LOAD RAM ADDRESS AGAIN
001020  022064  000006                  CMP     (R0)+,6(R4)             ;B-EQUAL TO WHAT WAS JUST WRITTEN?
001024  001062                          BNE     46$                     ;B-BR IF NOT, BAD ERROR
001026  005201                          INC     R1                      ;B-SET TO NEXT RAM ADDRESS
001030  005302                          DEC     R2                      ;B-ONE LESS WORD IN BUFFER
001032  003351                          BGT     20$                     ;B-BR IF MORE IN BUFFER
001034  012600                          MOV     (SP)+,R0                ;B-RESTORE BUFFER ADDRESS
001036  062767  000001  000002G        ADD     #1,$LBN+2               ;B-NEXT TEXT BLOCK IN FILE
001044  005567  000000G                 ADC     $LBN                    ;B-SET HIGH ORDER WORD
001050  005303                          DEC     R3                      ;B-ANY MORE BLOCKS TO READ?
001052  003333                          BGT     10$                     ;B-BR IF YES, KEEP GOING
                                        ;
                                        ; FINISHED FILE, CLEAN KMC, CLOSE FILE, EXIT WITH CARRY CLEAR
                                        ;
001054  010401                          MOV     R4,R1                   ;B-COPY KMC CSR ADDRESS
001056  005021                          CLR     (R1)+                   ;B-CLEAR ALL KMC REGISTERS AGAIN
001060  005021                          CLR     (R1)+
001062  005021                          CLR     (R1)+
001064  005011                          CLR     (R1)
001066                                  CALL    $CLDEQ                  ;B-CLOSE FILE
001072  000241                          CLC                             ;B-NO ERRORS
001074  000456                          BR      KMCEX                   ;B-EXIT
```

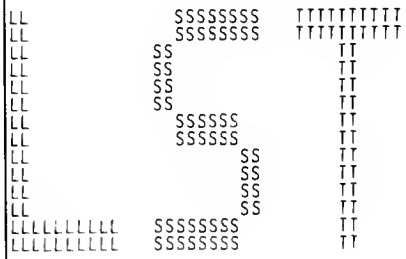
Table of contents

5-	54	MACRO DEFINITIONS
6-	96	LOCAL DATA
7-	256	$XFER - TRANSFER PROCESS TO CORE
8-	325	AXSYM - AUX SPECIAL SYMBOLS
9-	375	$CTSYM - FIND SYMBOLS FROM CETAB
10-	453	PRSYM - LOOK FOR PROCESS SYMBOLS
11-	586	$INICO - INITIALISE XPT COUNTERS
12-	609	$ALMHC - ALLOCATE MINHOPS/MINCOST MATRIX
13-	666	$ALECL - ALLOCATE ECL NODE COUNTERS

```
000724  000402                          BR      60$             ;* (REMEMBER - MAPX ADDS 1 WORD TO STACK)

000726  011101          50$:    MOV     (R1),R1         ;* GET NEXT SYMBOL BLOCK
000730  001365                  BNE     40$             ;* IF NE, NOT AT END YET
000732                  60$:
000732                          RETURN                  ;* BACK TO USER MODE

000734  005700          70$:    TST     R0              ; ANY PROBLEM?
000736  001004                  BNE     111$            ; IF NE, YES
000740                  80$:    RETURN

                        ;
                        ; ERROR CONDITIONS
                        ;
000742                  101$:   EMSG$R  49              ; DISPATCH TABLE SYMBOL UNDEFINED
000750                  111$:   EMSG$R  50              ; PROCESS/CEX INCOMPATIBLE
000756  016000  000162' STBERR: MOV     ETAB1-2(R0),R0  ; GET ERROR MESSAGE CONTROL BLOCK ADDRESS
000762                          CALLR   @(R0)+          ; PRINT ERROR MESSAGE
```

FILEID**NTLROO





NTLROO - NTL ROOT SEGMENT MACRO V05.03b Saturday 29-Jun-85 00:32

Table of contents

```
    546 000234  016701  000062'          60$:    MOV     $PDVA,R1        ; GET PDV ADDRESS
    547
    548 000240  032761  000000G 000000G          BIT     #ZF.INI,Z.FLG(R1) ; PROCESS NEED INITIALIZATION DISPATCH ?
    549 000246  001406                            BEQ     65$             ; IF NO - BRANCH
    550 000250  012767  000030  001264'          MOV     #FS.INI!FC.PCT,$$PDIN ; SET INIT FUNCTION CODE
    551 000256  004767  000000G                   CALL    $DSINI          ; IF YES - CALL PROCESS
    552 000262  103412                            BCS     101$            ; IF ERROR - BRANCH
    553
    554 000264  016701  000062'          65$:    MOV     $PDVA,R1        ; GET PDV ADDRESS
    555 000270  152761  000000G 000000G          BISB    #ZF.MFL,Z.FLG(R1) ; FORCE "CLEAR PROCESS" TO REMOVE FROM CORE
    556 000276  016761  000070' 000000G          MOV     $PCB,Z.PCB(R1)  ; INDICATE PROCESS IS CORE-RESIDENT
    557 000304  000241                            CLC
    558
    559 000306  000207                    70$:    RETURN
    560
    561                                   ;
    562                                   ; ERROR CONDITION
    563                                   ;
    564 000310  000167  000000G           101$:   JMP     $XPRO           ; DO A PARTIAL "CLEAR PROCESS"
    565
```



```
    566
    567                                   .END
```

NTLROO
SYMBOL
SYMBOL
L.NSTA
L.OWNR
L.UNT
MAXOV
MS.BUF
MS.CIR
MS.CSR
MS.EXT
MS.INC
MS.LIN
MS.LLC
MS.MTP
MS.MUX
MS.NT1
MS.NT2
MS.PAR
MS.PRI
MS.SEC
MS.VCT
NBIAS
NETPF
NONSI
NS0
NS1
NS2
NS3
NS4
NS5
NS6
NS7
NTLPT
NTUIC
OBJHD
PARHD
PAVL
PDVID
PDVNM
PDVTA
PLGTH
PSIPT
PUMR
P.LGTH
P.REL
QASTT
QBIAS
QSTRT
RDBAF
RDBCT
RDBLH
RDBNM
RDBSZ
RDBTH
NTLROO

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 00:32 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
L.NSTA	000014		#4-61			
L.OWNR	000021		#4-61			
L.UNT	000013		#4-61			
MAXOV	000102	RG	#5-143			
MS.BUF =	000100	G	#6-256			
MS.CIR =	010000	G	#6-263	14-830		
MS.CSR =	000001	G	#6-249			
MS.EXT =	002000	G	#6-261	11-674	11-691	14-818
MS.INC =	001000	G	#6-260	11-674	11-691	14-818
MS.LIN =	004037	G	#6-254			
MS.LLC =	000040	G	#6-255			
MS.MTP =	000010	G	#6-252			
MS.MUX =	004020	G	#6-253			
MS.NT1 =	000300	G	#6-258			
MS.NT2 =	000400	G	#6-259			
MS.PAR =	000200	G	#6-257			
MS.PRI =	000002	G	#6-250			
MS.SEC =	004000	G	#6-262			
MS.VCT =	000004	G	#6-251			
NBIAS	000100	RG	#5-142			
NETPF	000202	RG	#6-181			
NONSI	000046	RG	#5-119			
NS0	000156	RG	#6-171			
NS1	000160	RG	#6-172			
NS2	000162	RG	#6-173			
NS3	000164	RG	#6-174			
NS4	000166	RG	#6-175			
NS5	000170	RG	#6-176			
NS6	000172	RG	#6-177			
NS7	000174	RG	#6-178			
NTLPT	000232	RG	#6-193			
NTUIC	000074	RG	#5-137			
OBJHD	000234	RG	#6-194			
PARHD	000054	RG	#5-122			
PAVL	000244	RG	#6-202			
PDVID	000176	RG	#6-179			
PDVNM	000142	RG	#6-165			
PDVTA	000140	RG	#5-158			
PLGTH	000056	RG	#5-123			
PSIPT	000272	RG	#6-213			
PUMR	000236	RG	#6-195			
P.LGTH =	******	GX	5-123			
P.REL =	******	GX	12-752			
QASTT	000034	RG	#5-114			
QBIAS	000252	RG	#6-205			
QSTRT	000254	RG	#6-206			
RDBAF	000124	RG	#5-152			
RDBCT	000112	RG	#5-147			
RDBLH	000224	RG	#6-190			
RDBNM	000256	RG	#6-207			
RDBSZ	000216	RG	#6-187			
RDBTH	000104	RG	#5-144			

NTLRQB
LOCAL

12
12
12
12
12
12
12
12

NTLROO CREATED BY MACRO ON 29-JUN-85 AT 09:32 PAGE 4

NTLRQE
ERROR

LOCAL DATA

```
                                ;
                                ; REQUEST BLOCK TO $FLAGS BIT MAPPING TABLE
                                ;
000136  004000  000000G  LBITS:  .WORD   LS.FDX,FL.FDX   ; SET LINE - FDX, HDX, ECHO, TOP
000142  010000  000000G         .WORD   LS.HDX,FL.HDX
000146  001000  000000G  PBITS:  .WORD   LS.ECH,FL.ECH   ; SET PROCESS - ECHO, TOP
000152  002000  000000G         .WORD   LS.TOP,FL.TOP
000156  000000                  .WORD   0
```


ERROR MESSAGES

```
                                        .SBTTL  CLRPP - CLEAR PROCESS PARAMETERS
                                ;+
                                ; CLRPP - CLEAR PROCESS PARAMETERS
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       $$INC=0
                                ;       $$PAR=0
                                ;
                                ;-
001244  005067  000000G         CLRPP:  CLR     $$INC           ; SET NO PROCESS INCREMENT
001250  005067  000000G                 CLR     $$PAR           ; AND NO PARTITION OVERRIDE
001254                                  RETURN
```

```
                                        .SBTTL  $FSLT - FIND SLT

                                ;+
                                ;
                                ; $FSLT - TRY TO FIND SLT
                                ;
                                ; INPUTS:
                                ;       R1=CONTROLLER NUMBER (BYTE)
                                ;       R2=UNIT NUMBER (BYTE)
                                ;       PDVX=DDM PDV INDEX
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=SLT ADDRESS
                                ;       R1=SYSTEM LINE NUMBER
                                ;       R2=NUMBER OF UNITS ON THIS CONTROLLER
                                ;
                                ;-
003140                          $FSLT::
003140  012746  177777                  MOV     #-1,-(SP)       ; LEAVE ROOM FOR SLT ADDRESS AND SLN
003144  011646                          MOV     (SP),-(SP)      ;
003146  110146                          MOVB    R1,-(SP)        ; SAVE CONTROLLER NUMBER
003150  110266  000001                  MOVB    R2,1(SP)        ; AND UNIT NUMBER
003154  017700  000000G                 MOV     @SLTMA,R0       ; GET FIRST SLT ADDRESS
                                ;
003160  012701  177777                  MOV     #-1,R1          ; GET FIRST LINE NUMBER
003164  005067  000130'                 CLR     UNITS           ; NO UNITS THIS CONTROLLER YET
003170  012002                  10$:    MOV     (R0)+,R2        ; GET NEXT SLT ADDRESS
                                ;
003172  005201                          INC     R1              ; AND NEXT LINE NUMBER
003174  027701  000000G          20$:    CMP     @SLTNM,R1       ; IS IT END OF SLT?
003200  001436                          BEQ     30$             ; IF EQ, YES
003202  126267  000002  000126'         CMPB    L.DDM(R2),PDVX  ; DOES DDM PDV INDEX MATCH?
003210  001367                          BNE     10$             ; IF NE, NO
003212  126216  000012                  CMPB    L.CTL(R2),(SP)  ; DOES CONTROLLER NUMBER MATCH?
003216  001364                          BNE     10$             ; IF NE, NO
003220  126267  000013  000130'         CMPB    L.UNT(R2),UNITS ; ARE UNIT NUMBERS IN ASCENDING ORDER?
003226  001034                          BNE     10$             ; IF NE, NO
003230  016746  000130'                 MOV     UNITS,-(SP)     ; SAVE UNIT FOUND ON STACK
003234  062716  000002G                 ADD     #$SLN+2,(SP)    ; POINT TO SYSTEM LINE NUMBER
003240  110136                          MOVB    R1,@(SP)+       ; SAVE MUX LINE NUMBER
003242  005267  000130'                 INC     UNITS           ; TALLY UNITS ON THIS CONTROLLER
003246  126266  000013  000001         CMPB    L.UNT(R2),1(SP) ; DOES UNIT NUMBER MATCH?
003254  001345                          BNE     10$             ; IF NE, NO
003256  010266  000002                  MOV     R2,2(SP)        ; SAVE SLT ADDRESS
003262  010166  000004                  MOV     R1,4(SP)        ; SAVE SYSTEM LINE NUMBER
003266  032767  000000G 000000G         BIT     #FL.MUX,$FLAGS  ; MUX DEVICE?
003274  001335                          BNE     10$             ; IF NE, YES - SEARCH ENTIRE SLT
003276  005726                  30$:    TST     (SP)+           ; PURGE STACK (CLEARS C-BIT)
003300  012600                          MOV     (SP)+,R0        ; RETRIVE SLT ADDRESS
003302  012601                          MOV     (SP)+,R1        ; SYSTEM LINE NUMBER
003304  100403                          BMI     35$             ; IF MI, LINE NOT FOUND
003306  016702  000130'                 MOV     UNITS,R2        ; RETURN UNITS FOUND IN R2
003312  001001                          BNE     40$             ; IF NE, SUCCESS
003314  000261                  35$:    SEC                     ; INDICATE FAILURE
003316                          40$:    RETURN
```

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
$C5TA	= ****** GX	28-1287	30-1351							
$DDMPV	= ****** GX	*14-448	14-478							
$DLCPV	= ****** GX	*14-481								
$ERRZP	001406 R	#7-178	15-590							
$ERRZR	001264 R	#7-175	18-672							
$ERRZ1	001120 R	#7-167	15-589							
$ERRZ3	001150 R	#7-168	18-713	18-718	21-942					
$ERRZ4	001162 R	#7-169	18-757	21-980						
$ERRZ5	001330 R	#7-176	18-776	21-1001						
$ERRZ6	001354 R	#7-177	19-809							
$ERRZ7	001224 R	#7-172	18-754	21-977						
$ERR86	001062 R	#7-161	22-1055							
$ERR87	001014 R	#7-160	25-1187							
$ERR88	000752 R	#7-159	14-489							
$ERR90	000716 R	#7-158	14-488							
$ERR91	000660 R	#7-154	9-284							
$ERR92	000616 R	#7-152	15-588							
$ERR93	000552 R	#7-151	15-587							
$ERR94	000516 R	#7-150	15-586							
$ERR95	000462 R	#7-149	14-487							
$ERR96	000424 R	#7-148	11-346							
$ERR97	000366 R	#7-147	11-345							
$ERR98	000330 R	#7-146	10-310							
$ERR99	000276 R	#7-145	8-241							
$FLAGS	= ****** GX	*9-260	*9-280	*11-339	12-377	*14-451	*14-455	*14-459	18-674	18-761
		*22-1038	23-1074	25-1172						
$FPDV	003050 RG	11-333	14-443	22-1033	#24-1101					
$FSLT	003140 RG	14-467	22-1042	#25-1144						
$FUNC	= ****** GX	*8-235								
$MAPX	= ****** GX	20-837								
$MISS	= ****** GX	*8-230	*12-369	*12-375	*12-381	*13-409	*15-516	*15-536	*15-542	*15-551
		*15-569	*22-1050							
$MXVCT	= ****** GX	15-549								
$NAME	= ****** GX	*11-332	*14-442	18-661	*22-1032					
$NLCSR	= ****** GX	15-534	15-567							
$NLRQB	000000 RG	#8-218								
$PCB	= ****** GX	*13-412								
$PDVA	= ****** GX	*11-335	13-410	*14-445	18-654	18-727	*21-913	*22-1035	28-1271	28-1285
$RERR	003332 RG	7-145	7-146	7-147	7-148	7-149	7-150	7-151	7-152	7-154
		7-158	7-159	7-160	7-161	7-167	7-168	7-169	7-172	7-175
		7-176	7-177	7-178	#26-1204					
$SAVAL	= ****** GX	26-1205								
$SAVRG	= ****** GX	21-885								
$SAVVR	= ****** GX	18-690	21-931							
$SLN	= ****** GX	*14-470	*22-1045	25-1165						
$SLTA	= ****** GX	*14-469	17-627	19-795	*22-1044					
$TIOUT	= ****** GX	26-1222								
$TRIB	= ****** GX	*22-1049								
$$CSR	= ****** GX	*15-537								
$$DEV	= ****** GX	*12-385								
$$EXT	= ****** GX	*12-376	*12-379							
$$HTIM	= ****** GX	15-574								

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 6

```
                                ;+
                                ; $NLCDD - CALL DEVICE DRIVER AT POWERFAIL ENTRY (MAPPED 11M ONLY)
                                ;
                                ; THIS ROUTINE IS USED TO CALL THE LLC'S DEVICE DRIVER AT HIS POWERFAIL
                                ; ENTRY POINT FOR ALL UNITS WHICH HAVE THE UC.PWF BIT SET IN U.CTL.
                                ;
                                ; INPUTS:
                                ;       R3=DCB ADDRESS
                                ;       R4=NUMBER OF UNITS TO SCAN
                                ;       R5=FIRST UCB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-

                                        .IF DF  R$$11M & M$$MGE

000352                          $NLCDD::
                                        SWSTK$  30$             ;* ENTER KERNEL MODE
000352  010446                          MOV     R4,-(SP)        ;* SAVE UNIT COUNT
000356  132765  000000G 000000G 10$:    BITB    #UC.PWF,U.CTL(R5) ;* CALL DRIVER AT POWERFAIL ENTRY?
000360  001413                          BEQ     20$             ;* IF EQ, NO
000366  010346                          MOV     R3,-(SP)        ;* SAVE DCB ADDRESS
000370  010546                          MOV     R5,-(SP)        ;* AND UCB ADDRESS
000372  016302  000000G                 MOV     D.DSP(R3),R2    ;* GET DISPATCH TABLE ADDRESS
000376  016504  000000G                 MOV     U.SCB(R5),R4    ;* AND SCB ADDRESS
000402                                  CALL    @LDPWF          ;* CALL DRIVER AT POWERFAIL ENTRY

                                        .IF NDF R$$MPL
000406  005726                          TST     (SP)+           ;* CLEAN UP AFTER ANDTHER RSX11M HACK!!!
                                        .ENDC

000410  012605                          MOV     (SP)+,R5        ;* RESTORE UCB ADDRESS
000412  012603                          MOV     (SP)+,R3        ;* AND DCB ADDRESS
000414  016765  000000G 000000G 20$:    MOV     $FEATR,U.CW3(R5) ;* SETUP THE FEATURES WORD
000422  066305  000000G                 ADD     D.UCBL(R3),R5   ;* GIVING NEXT UCB ADDRESS
000426  005316                          DEC     (SP)            ;* ALL DONE?
000430  001352                          BNE     10$             ;* IF NE, NO
000432  005726                          TST     (SP)+           ;* PURGE STACK
000434                          30$:    RETURN                  ;* BACK TO USER MODE

                                        .ENDC
```

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES
.CXLBR	000060	G	#5-150
.CXLB1	000030	G	#5-148
.CXLB2	000044	G	#5-149
.CXLEN	000064	G	#5-152
.CXPCB	000006	G	#5-143
.CXSYM	000010	G	#5-144
.CXUNL	000024	G	#5-146
.CXVEC	000026	G	#5-147
.CXVSM	000016	G	#5-145
.DPRB	000041	G	#5-110
.DPRW	000042	G	#5-111
.DVCHA	000011	G	#5-86
.EOF	000012	G	#5-87
.FILE	000013	G	#5-88
.INT	000014	G	#5-89
.LFILE	000017	G	#5-92
.LIBR	000015	G	#5-90
.LINKS	000016	G	#5-91
.LSTHD	000020	G	#5-93
.LTAB	000021	G	#5-94
.MPTAB	000022	G	#5-95
.MXPTB	000044	G	#5-113
.MXTAB	000023	G	#5-96
.NODE	000024	G	#5-97
.PECHA	000045	G	#5-114
.PRI	000025	G	#5-98
.PRTOV	000026	G	#5-99
.SCOM	000027	G	#5-100
.SECSR	000030	G	#5-101
.SLNB	000031	G	#5-102
.SLNW	000032	G	#5-103
.STNB	000033	G	#5-104
.STNW	000034	G	#5-105
.TIME	000043	G	#5-112
.UNB	000035	G	#5-106
.UNW	000036	G	#5-107
.WORD	000037	G	#5-108
.X2CH1	000061	G	#5-126
.X2CH2	000062	G	#5-127
.X2CH3	000063	G	#5-128
.X2CH4	000064	G	#5-129
.X2CH5	000065	G	#5-130
.X3C01	000046	G	#5-115
.X3C02	000047	G	#5-116
.X3C03	000050	G	#5-117
.X3C04	000051	G	#5-118
.X3C05	000052	G	#5-119
.X3C06	000053	G	#5-120
.X3C07	000054	G	#5-121
.X3C08	000055	G	#5-122
.X3C09	000056	G	#5-123
.X3C10	000057	G	#5-124

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 3

```
                                ;+
                                ; $EVEN - ".EVEN" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000470                          $EVEN::
000470  032767  000001  000014' BIT     #1,$SIZE        ; IS SIZE CURRENTLY EVEN?
000476  001410                  BEQ     10$             ; IF EQ, YES - DO NOTHING
000500                          CALL    $RQ2            ; REQUEST BINARY FILE
000504  103405                  BCS     10$             ; IF CS, NO ACCESS
000506  112722  000000G         MOVB    #.BYTE,(R2)+    ; STORE FUNCTION CODE
000512  105012                  CLRB    (R2)            ; STORE ZERO VALUE
000514  005267  000014'         INC     $SIZE           ; ADJUST DATA BASE SIZE
000520                  10$:    RETURN
```

```
                                .SBTTL  MACRO DEFINITIONS

                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  ASR$,OBJDF$,NKRDF$,RNBDF$,ANBDF$,FLTDF$,MRKT$S,WTSE$S,DHBDF$
                                .MCALL  EVLDF$,SLTDF$,ASL$,SAVRG,RESRG,DECL$S,XPDDB$,CTRDF$,ECDDB$
                                .MCALL  SERDF$,FNBDF$,CCBDF$,WSIG$S,ADJDF$,CALLR,STMDF$

000000                          ADJDF$                  ; DEFINE ADJACENT DATA BASE OFFSETS
000000                          ANBDF$                  ; DEFINE ALIAS NAME BLOCK OFFSETS
000000                          CTRDF$                  ; DEFINE COUNTER BLOCK OFFSETS
000000                          ECDDB$                  ; DEFINE ECL DDB OFFSETS
000000                          EVLDF$                  ; DEFINE EVENT LOGGING OFFSETS
000000                          FLTDF$                  ; DEFINE FILTER CONTROL BLOCK OFFSETS
000000                          NKRDF$                  ; DEFINE NETWORK KRB OFFSETS
000000                          XPDDB$                  ; DEFINE XPT DDB OFFSETS
000000                          OBJDF$                  ; DEFINE OBJECT TABLE OFFSETS
000000                          RNBDF$                  ; DEFINE REMOTE NAME BLOCK OFFSETS
000000                          SLTDF$                  ; DEFINE SLT SYMBOLS AND OFFSETS
000000                          DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS
000000                          SERDF$                  ; DEFINE SERVER BLOCK OFFSETS
000000                          FNBDF$                  ; DEFINE FILENAME BLOCK OFFSETS
000000                          CCBDF$                  ; DEFINE CCB FUNCTION CODE SYMBOLS
000000                          STMDF$                  ; DEFINE SHORT TIMER BLOCK OFFSETS


                                .MCALL  TCBDF$
000000                          TCBDF$                  ; DEFINE TCB OFFSETS

                                .IF DF  R$$MPL

                                .MCALL  KRBDF$,CTBDF$,STAF$,RMAF$S,DIR$

                                KRBDF$                  ; DEFINE KRB OFFSETS
                                CTBDF$                  ; DEFINE CTB OFFSETS
                        SETAF:  STAF$   0               ; SET AFFINITY

                                .ENDC


                        ;
                        ; LOCAL MACRO DEFINITIONS
                        ;
                                .MACRO  SAVMAP
                                MOV     @KSAR5,-(SP)    ; SAVE APR 5
                                .ENDM

                                .MACRO  RESMAP
                                MOV     (SP)+,@KSAR5    ; RESTORE APR5
                                .ENDM
```

```
                                ;+
                                ; $NLGET - RECEIVE DATA FROM CALLING TASK
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       $$LRB=REQUEST BLOCK (FIRST 2 WORDS=CALLING TASK'S NAME)
                                ;-
000146                          $NLGET::
000146                                  DIR$    #RCVDAT         ; RECEIVE DATA OR EXIT FROM TASK
000154  016767  000022' 000000G         MOV     $$LRB,$$TASK    ; COPY REQUESTING TASK'S NAME
000162  016767  000024' 000002G         MOV     $$LRB+2,$$TASK+2 ;      ...
000170  016767  000002G 000000G         MOV     $FLAGS+2,$FLAGS ; RESET GLOBAL NETLDR STATUS
                                        .IF NDF I$$AS
000176  005067  000000G                 CLR     $$AST           ; ASSUME CALLER NOT NOTIFIED VIA AST
000202  032767  100000  000026'        BIT     #LS.AST,$$LRB+4 ; NOTIFY CALLER VIA AST?
000210  001406                        BEQ     10$             ; BR IF NO
000212  016767  000054' 000000G         MOV     LR.AST+4+$$LRB,$$AST ; SAVE NOTIFICATION AST ADDRESS
000220  016767  000056' 000000G         MOV     LR.PRM+4+$$LRB,$$PRM ; SAVE NOTIFICATION AST PARAMETER TO RETURN
                                        .ENDC
000226  042767  000000C 000000G 10$:    BIC     #^C<FL.ECH!FL.TOP!FL.WRN>,$FLAGS ;      ...
000234                          RTSPC:  RETURN
```

```
                                        .SBTTL  ERROR PROCESSING DATA AND FILE BUFFER


                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
                                ; ALL ERRORS ARE PREFIXED WITH 'NTL -- ''
                                ;
000024     015     116     124  KMCEBF: .ASCII  <15>/NTL -- /
000034                          KMCEMS: .BLKB   70.
                                        .EVEN


                                ;
                                ; ERROR MESSAGE CONTROL MACROS
                                ;
                                ; ERROR PREFIXS (FORMAT STRINGS) ARE:
                                ;       10A - ''*FILNAM* MICRO-CODE FILE -- ''
                                ;       10B - ''*KMCDEV* DEVICE I-O ERROR, CSR = *ADDR*, RAM ADDRESS = *ADDR*''
                                ;       10C - ''LIST ALLOC. FAILURE, DEV = *DEV*, CSR = *ADDR*, CNTRLR = *NUM*''
                                ;
                                ; ERROR CONTROL ROUTINES ARE:
                                ;       KMCEFT - ERROR FORMATTING MAIN CONTROL
                                ;       KMCFEC - FILNAM IN PREFIX / ERRCODE (MAYBE) IN SUFFIX / FILE CLOSED
                                ;       KMCFEO - FILNAM IN PREFIX / ERRCODE (MAYBE) IN SUFFIX / FILE OPEN
                                ;       KMCRL1 - KMCDEV, CSR AND RAM ADDRESS IN PREFIX / SUFFIX NOT USED
                                ;       KMCRL2 - VALUE WRITTEN AND VALUE READ IN SUFFIX / PREFIX NOT USED
                                ;
                                        .ENABL  LC
000142                          NTLER$  ,3A,10A,KMCEFT,KMCFEC,,<Open Failure (*.)>
000174                          NTLER$  ,3C,10A,KMCEFT,KMCFEO,,<File Not Contiguous>
000230                          NTLER$  ,3D,10A,KMCEFT,KMCFEO,,<Label Block Read Failure (*.)>
000276                          NTLER$  ,3E,10A,KMCEFT,KMCFEO,,<File Contains a Task Header>
000342                          NTLER$  ,3F,10A,KMCEFT,KMCFEO,,<File Is An Overlaid Task>
000404                          NTLER$  ,3H,10A,KMCEFT,KMCFEO,,<Read Failure (*.)>
000436                          NTLER$  ,31,10B,KMCEFT,KMCRL1,KMCRL2,<* Was Written, * Was Read>
000500                          NTLER$  ,3J,10C,KMCEFT,KMCRL1,,
                                        .IF DF  R$$MPL
                                NTLER$  ,3K,10D,KMCEFT,KMCFEC,,<Unable to find Unibus Run Mask>
                                        .ENDC
                                        .DSABL  LC


                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000512     052     040     115  FMT10A: .ASCIZ  ''* Micro-Code File -- ''
000540     052     040     104  FMT10B: .ASCIZ  ''* Device I/O Error, CSR = *, RAM Address = *''
000615     114     151     163  FMT10C: .ASCIZ  ''List Alloc. Failure, Dev = *, CSR = *, Cntrlr = *''
                                        .IF DF  R$$MPL
                                FMT10D: .ASCIZ  ''* Micro-Code Load -- ''
                                        .ENDC
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX
```

```
                                        ;
                                        ; ERROR PROCESSING
                                        ;
001076                          30$:    EMSG$   3A                      ;B-OPEN FAIL:   FILNAM+ERROR
001104  000451                         BR      55$
                                
001106                          34$:    EMSG$   3C                      ;B-NOT CONTIG:  FILNAM
001114  000443                         BR      50$
                                
001116  116777  000000G 000000G 36$:    MOVB    $IOSB,@$CLIOS           ;B-RD LABEL FL: FILNAM+ERROR
001124                                  EMSG$   3D
001132  000434                         BR      50$
                                
001134                          38$:    EMSG$   3E                      ;B-HEADER:      FILNAM
001142  000430                         BR      50$
                                
001144                          40$:    EMSG$   3F                      ;B-FIL OVRLAID: FILNAM
001152  000424                         BR      50$
                                
001154  116777  000000G 000000G 44$:    MOVB    $IOSB,@$CLIOS           ;B-READ ERROR:  FILNAM+ERROR
001162                                  EMSG$   3H
001170  000415                         BR      50$
                                
001172  010467  000002'          46$:    MOV     R4,KMCCSR               ;B-RAM ERROR:   DEV, CSR, ADDR, I/O
001176  010167  0[illegible]'            MOV     R1,KMCER1               ;B-SAVE THE CSR AND RAM ADDRESSES
001202  014067  C[illegible]'            MOV     -(R0),KMCER2            ;B-ALSO THE VALUE WRITTEN TO THE RAM
001206  016467  0[illegible]  000022'    MOV     6(R4),KMCER3            ;B-AND THE VALUE READ FROM THE RAM
001214  005726                         TST     (SP)+                   ;B-DUMP THE SAVED BUFFER ADDRESS
001216                                  EMSG$   3I                      ;B-AND GIVE THE MESSAGE
                                
001224                          50$:    CALL    $CLDEQ                  ;B-CLOSE FILE
001230  000261                  55$:    SEC                             ;B-INDICATE ERROR
                                        ;
                                        ; EXIT KMC LOADER CODE
                                        ;
001232                          KMCEX:
                                        .IF DF  R$$MPL
                                
                                        TSTB    $NCPU                   ;M+ IS THIS A MULTI-PROCESSOR SYSTEM ?
                                        BEQ     60$                     ;M+ IF EQ, NO
                                        RMAF$S                          ;M+ REMOVE THE CPU AFFINITY
                                60$:                                    ;M+
                                        .ENDC
                                
001232                                  RETURN
```

```
                        .TITLE  NTLPRO - ''SET PROCESS'' ROUTINES
                        .IDENT  /V05.00/

;
;
;
```

```
                                        .SBTTL  $INICO - INITIALISE XPT COUNTERS
                                ;+
                                ; $INICO - INITIALISE XPT COUNTERS AND SET UP ROL TABLE AND NI CACHE
                                ;          AND ADJACENCY DATA BASE
                                ;
                                ; INPUTS:
                                ;       $PDVA - XPT'S PDV ADDRESS
                                ;
                                ; REGISTERS MODIFIED:
                                ;       NONE
                                ;-
000764  016705  000000G         $INICO::MOV     $PDVA,R5        ; COPY XPT PDV ADDRESS
000770  016502  000000G                 MOV     Z.DAT(R5),R2    ; STORE XPT DDB ADDRESS
000774                                  CALL    $NICAH          ; ALLOCATE NI CACHE
001000                                  CALL    $ADJ            ; ALLOCATE ADJACENCY DATA BASE
001004  103405                          BCS     10$             ; IF FAILURE - BRANCH
001006                                  CALL    $ROL            ; ALLOCATE REACHIBILITY MATRIX
001012  103402                          BCS     10$             ; IF FAILURE - BRANCH
001014                                  CALL    $CKT            ; ALLOCATE CIRCUIT COUNTERS

001020                          10$:    RETURN
```

Table of contents

4-	54	MACRO DEFINITIONS
5-	69	VECTORED EXEC TABLE
7-	437	$NTL - NETLDR ENTRY POINT
8-	489	$SPRO - SET PROCESS
9-	568	VECSET - SET UP PROCESS VECTOR TABLE
10-	608	$VCTLB - SET UP LIBRARY VECTOR TABLE
11-	650	$SLIN - SET LINE
12-	718	AIC - TEST FOR ALREADY IN CORE
13-	759	$SCEX - SET CEX
14-	801	$SCIR - SET CIRCUIT

```
                        .SBTTL  VECSET - SET UP PROCESS VECTOR TABLE
                ;+
                ; VECSET - SET UP PROCESS VECTOR TABLE
                ;
                ; INPUTS:
                ;       $VCLN - NUMBER OF SYMBOLS
                ;       $VCTB - ADDRESS OF TABLE
                ;       $BIAS - BIAS OF PROCESS
                ;
                ; OUTPUTS:
                ;       VECTOR TABLE TRANSLATED
                ;-
                        .IF DF  R$$MPL

                VECSET: MOV     $VCLN,R2        ; COPY NUMBER OF SYMBOLS VECTORED
                        BEQ     10$             ; IF PROCESS NOT VECTORED - BRANCH
                        EMT     376             ; SWSTK$ 10$ WITHOUT RSXMC
                        .WORD   10$             ;
                        MOV     @KSAR6,-(SP)    ; SAVE OUR APR6 MAPPING
                        MOV     MPPRO,R0        ; GET ADDRESS OF PROCESS MAPPING ROUTINE
                        MOV     KSAR6,R5        ; GET KISAR6 ADDRESS
                        MOV     $BIAS,@KSAR6    ; MAP TO PROCESS
                        MOV     $VCTB,R3        ; COPY TABLE ADDRESS OF PROCESS

                        MOV     @#112,R1        ; COPY BIAS OF EXEC ROUTINE WHICH
                                                ; CALLS THE VECTOR TRANSLATE ROUTINE
                        MOV     (R1),R1         ; GETTING TO ACTUAL BIAS
                        MOV     (R1),-(SP)      ; PUT BIAS ON STACK FOR $MPPRO
                        MOV     #120000,-(SP)   ; MAP THRU APR5
                        CALL    (R0)            ; CALL DIRECTIVE COMMON ROUTINE THE LONG WAY
                        MOV     (SP)+,(R5)      ; RESTORE OUR APR6 MAPPING
                        CLR     $VCLN           ; INIT VECTOR TABLE FLAG

                10$:    RETURN

                        .ENDC
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
R$$MPL	= ******	4-64 5-92 5-129 6-197 6-230 6-345 7-456 8-512 9-582 10-622 11-681 11-698
R$$11D	= ******	12-741
SDBAF	000130 RG	#5-154
SDBCT	000110 RG	#5-146
SDBLH	000226 RG	#6-191
SDBNM	000260 RG	#6-208
SDBSZ	000262 RG	#6-209
SF.ACT	= 000200	#4-61
SF.ENA	= 000100	#4-61
SF.LPB	= 000004	#4-61
SF.MFL	= 000040	#4-61
SF.PAC	= 000020	#4-61
SF.REA	= 000010	#4-61
SF.SER	= 000001	#4-61
SF.SVC	= 000002	#4-61
SF.UNL	= 000040	#4-61
SLHD	000070 RG	#5-135
SLTMA	000144 RG	#6-166
SLTNM	000146 RG	#6-167
SRSTD	000032 RG	#5-113
STMFC	000242 RG	#6-201
SYSNM	000002 RG	#5-85
S$CEX	= ****** GX	13-776
S.COST	000001	#4-61
S.FLG	000000	#4-61
S.LEN	000004	#4-61
S.LHD	= ****** GX	5-135
S.NMST	000002	#4-61
S.OWNR	000003	#4-61
TKPS	000004 RG	#5-86
TKTCB	000072 RG	#5-136
TK100	000134 RG	#5-156
TTNS	000030 RG	#5-112
T1SCL	000214 RG	#6-186
T100C	000212 RG	#6-185
T100Q	000210 RG	#6-184
UISAR0	= ****** GX	5-124
UISAR7	= ****** GX	5-117
UISDR7	= ****** GX	5-118
USAR0	000060 RG	#5-124
USAR7	000042 RG	#5-117
USDR7	000044 RG	#5-118
V$$CTR	= ****** GX	5-138 6-427
XAVL	000250 RG	#6-204
XBIAS	000246 RG	#6-203
ZF.INI	= ****** GX	8-548
ZF.MFL	= ****** GX	8-555
ZF.PSE	= ****** GX	8-519
ZTIME	000064 RG	#5-126
Z.FLG	= ****** GX	8-519 8-548 *8-555 12-735
Z.NAM	= ****** GX	8-528 8-536 12-739

```
                                        .SBTTL  ERROR MESSAGES

                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
000160  015                     ERRMSG: .BYTE   15
000161  116     124     114             .ASCII  'NTL -- '
000170                          ERRBUF: .BLKB   70.

                                        .EVEN

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000276                                  NTLER$  ,99,7,$RERR,REP7,,<Illegal Function>
000330                                  NTLER$  ,98,7,$RERR,REP7,,<Comm Exec Not Loaded>
000366                                  NTLER$  ,97,7,$RERR,REP7,,<Process Not In System>
000424                                  NTLER$  ,96,7,$RERR,REP7,,<Garbage In Flags Byte>
000462                                  NTLER$  ,95,7,$RERR,REP7,,<Line Not In Syst.>
000516                                  NTLER$  ,94,7,$RERR,REP7,,<Illegal CSR Addr.s>
000552                                  NTLER$  ,93,7,$RERR,REP7,,<Illegal Interrupt Priority>
000616                                  NTLER$  ,92,7,$RERR,REP7,,<Illegal Interrupt Vector>
                                        .IF DF  R$$11M
000660                                  NTLER$  ,91,7,$RERR,REP7,,<No Data Block Address>
                                        .IFF
                                        NTLER$  ,91,7,$RERR,REP7,,<Data Block Allocation Failure>
                                        .ENDC
000716                                  NTLER$  ,90,7,$RERR,REP7,,<Illegal Unit Number>
000752                                  NTLER$  ,88,7,$RERR,REP7,,<Garbage In DLC Flags Byte>
001014                                  NTLER$  ,87,7,$RERR,REP7,,<Units Not In Ascending Order>
001062                                  NTLER$  ,86,7,$RERR,REP7,,<Circuit not in system>

                                        .IF DF  R$$MPL
                                        NTLER$  ,85,7,$RERR REP7,,<Comm exec has been deallocated>
                                        .ENDC

001120                                  NTLER$  ,Z1,7,$RERR,REP7,,<Device Offline>
001150                                  NTLER$  ,Z3,7A,$RERR,REP7A,,< >
001162                                  NTLER$  ,Z4,7,$RERR,DEV7,,<Device **: Still Mounted>

                                        .IF DF  R$$11M
001224                                  NTLER$  ,Z7,7,$RERR,DEV7,,<Device **: Still Active>
                                        .ENDC

001264                                  NTLER$  ,ZR,7,$RERR,REP7,,<Process May Not Be Unloaded>
001330                                  NTLER$  ,Z5,7B,$RERR,REP7B,,<Still Ready>
001354                                  NTLER$  ,Z6,7,$RERR,REP7,,<Line Still Active>
001406                                  NTLER$  ,ZP,7,$RERR,REP7,,<Vector Not In System>

                                        .DSABL  LC


                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
```

```
                                        .SBTTL  SLIN - CHECK SET LINE REQUEST BLOCK

                                ;+
                                ;
                                ; SLIN - CHECK "SET LINE" REQUEST BLOCK
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R4 CLEARED IF NO WORK TO DO
                                ;
                                ;-

001256  012700  000136'         SLIN:   MOV     #LBITS,R0       ; DO BIT TWIDDLING
001262                                  CALL    $BITS           ;
001266  016700  000000G                 MOV     $SLTA,R0        ; GET SLT ADDRESS
001272  032760  040000  000000          BIT     #LF.RDY,L.FLG(R0) ; IS LINE READY ?
001300  001401                          BEQ     10$             ; NO ..
001302  005004                          CLR     R4              ; YES .. NO WORK TO DO!
001304                          10$:    RETURN
```

```
                                        ;
                                        ; ERROR CONDITION
                                        ;
003320  016706  000132'         101$:   MOV     SPSAV,SP        ; RESTORE SP TO $NLRQB ENTRY VALUE
003324                                  EMSG$R  87              ; UNITS OUT OF ORDER
```

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 6

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$$INC	= ****** GX	*12-373 *16-607
$$LINX	= ****** GX	*12-387
$$LRB	= ****** GX	8-220 9-278 12-370 15-517 22-1028 23-1073
$$LTIM	= ****** GX	15-575
$$MTP	= ****** GX	*15-571
$$MUX	= ****** GX	*14-471 *22-1046
$$NODC	= ****** GX	*12-388
$$PAR	= ****** GX	*12-383 *12-384 *16-608
$$PRI	= ****** GX	*15-544
$$SNUM	= ****** GX	*15-572
$$TBL	= ****** GX	*12-386
$$VECT	= ****** GX	*15-552
.CBIAS	= ****** GX	14-474 30-1348
.CCEX	000054 R	6-80 #6-101
.CLIN	000072 R	6-82 #6-106
.CPRO	000062 R	6-81 #6-103
.CXCSR	= ****** GX	20-838
.HOME	= ****** GX	9-274 *9-282
.KMCPF	000102 R	6-83 #6-109
.SCEX	000020 R	6-77 #6-90
.SCIR	000110 R	6-84 #6-111
.SLIN	000040 R	6-79 #6-96
.SPRO	000026 R	6-78 #6-92
..SCSR	= ****** GX	*15-570

NTLRQB CREATED BY MACRO ON 29-JUN-85 AT 00:33 PAGE 7

```
                        ;+
                        ; *** - $HRCIN - REINITIALIZE HRC
                        ;
                        ; THIS SUBROUTINE IS USED AS A GENERALIZED INTERFACE TO REQUEST HRC...
                        ; TO REINITIALIZE ITSELF AFTER THE I/O DATA BASE HAS BEEN CHANGED.
                        ;
                        ; INPUT:
                        ;       NONE.
                        ;
                        ; OUTPUT:
                        ;       C-BIT = SUCCESS/FAILURE
                        ;
                        ;-
                        
                                .IF DF  R$$MPL
                        $HRCIN::
                                ALUN$S  #3,#''RD,#0             ; ASSIGN LUN TO RDO:
                                BCS     20$                     ; IF CS, EXIT
                        
                                QIOW$S  #IO.ATT,#3,#32.,,#STATUS  ; ATTACH TO THE RDO:
                                BCS     20$                     ; IF CS, ERROR
                                QIOW$S  #IO.HRC,#3,#32.,,#STATUS,,<#M$INIT,#M$INIT>
                                                                ; REQUEST HRC... TO
                                                                ; REINITIALIZE ITSELF
                                ROR     R1                      ; SAVE ERROR STATUS
                                QIOW$S  #IO.DET,#3,#32.         ; DETACH FROM RECONFIGURATION DRIVER
                                ROL     R1                      ; RESTORE CARRY
                                BCS     20$                     ; IF CS, EXIT
                                TSTB    STATUS                  ; WAS THE QIO SUCCESSFUL ?
                                BPL     20$                     ; IF PL, YES
                        10$:    SEC                             ; ELSE, INDICATE ERROR
                        20$:    RETURN
                                .ENDC
                        
                        000001          .END
```


Symbol table

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES
.X3C11	000060	G	#5-125

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 4

```
                                ;+
                                ; $VFDDM - VERIFY DDM ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ERROR UNLESS DDM BEING LOADED
                                ;-
000522                          $VFDDM::
000522  032767  000000G 000000G         BIT     #FL.DDM,$FLAGS  ; IS THIS A DDM?
000530  001003                         BNE     10$             ; IF NE, YES
000532                                  EMSG$   21              ; DDM OPERATOR IN DLC/LLC TEMPLATE
000540                          10$:    RETURN
```

```
                                .SBTTL  LOCAL SYMBOL DEFINITIONS


                        ;
                        ; SAVED REGISTER OFFSETS FOR SWSTK$
                        ;
000002                  R$R0 = 2
000004                  R$R1 = 4
000006                  R$R2 = 6
000010                  R$R3 = 10
000012                  R$R4 = 12
000014                  R$R5 = 14


                        ;
                        ; PROCESS INIT/TERM FLAG
                        ;

000001                  INIFL = 1
```

```
                                ;+
                                ; $NLPUT - SEND BACK 1 WORD SUCCESS/FAILURE CODE TO CALLER
                                ;
                                ; INPUTS:
                                ;       $$TASK=CALLING TASK'S NAME
                                ;       $FUNC=SUCCESS CODE
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000236                          $NLPUT::
                                        .IF NE I$$AS
000236  005767  000000G                 TST     $$AST           ; NOTIFY USER VIA AST?
000242  001403                          BEQ     5$              ; BR IF YES
000244                                  CALL    PUTAST          ; PUT COMPLETION AST IN CALLER'S AST QUEUE
000250  000422                          BR      20$             ; AND EXIT
                                        .ENDC
000252  016767  000000G 000012' 5$:     MOV     $$TASK,ANSWER+S.DATN ; COPY CALLING TASK'S NAME
000260  016767  000002G 000014'         MOV     $$TASK+2,ANSWER+S.DATN+2 ;      ...
000266  012700  000026'                 MOV     #$$LRB+4,R0     ; GET REQUEST BLOCK ADDRESS
000272  016720  000000G                 MOV     $FUNC,(R0)+     ; COPY FUNCTION CODE
000276  012701  000014                  MOV     #12.,R1         ; ZERO OUT THE NEXT 12. WORDS
000302  005020                  10$:     CLR     (R0)+           ;       ...
000304  005301                          DEC     R1              ;       ...
000306  001375                          BNE     10$             ;       ...
000310                                  DIR$    #ANSWER         ; SEND DATA BACK TO CALLER
000316                          20$:    RETURN
```

```
                                ;
                                ; BUFFER FOR USE IN FILE I/O, MUST RESIDE AT END OF OVERLAY TO TAKE
                                ; ADVANTAGE OF EXTRA ROOM, SO USE APPROPRIATE PSECT NAME
                                ;
000000                                  .PSECT  ..KMC,D
000000                          KMCBUF: .BLKW   1                       ;USE EXTRA ROOM AT END


                                ;
                                ; START OF INSTRUCTION PSECT, MUST BE PURE CODE
                                ;
000000                                  .PSECT
```

```
                                        .SBTTL  MAIN ERROR PROCESSING


                                ;+
                                ; -KMCEFT- MAIN ERROR FORMATTER
                                ;
                                ; THIS ROUTINE IS RESPONSIBLE FOR CALLING PROPER ERROR FORMATTERS
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS OF MESSAGE CONTROL BLOCK PLUS TWO
                                ;       MESSAGE CONTROL BLOCK FORMAT:
                                ;               WORD    - ADDRESS OF MAIN ERROR PROCESSOR
                                ;               WORD    - ADDRESS OF ERROR MESSAGE PREFIX (FORMAT STRING)
                                ;               WORD    - ADDRESS OF FIRST MESSAGE FORMATTING ROUTINE
                                ;               WORD    - ADDRESS OF SECOND FORMATTING ROUTINE (OPTIONAL)
                                ;               ASCIZ   - ERROR MESSAGE SUFFIX CHARACTER STRING
                                ;
                                ; OUTPUT:
                                ;       PROPER MESSAGES ARE FORMATTED AND PRINTED
                                ;       R0, R1, R2, AND R3 ARE DESTROYED
                                ;-

001234  010446                  KMCEFT: MOV     R4,-(SP)                ;SAVE R4
001236  010546                          MOV     R5,-(SP)                ;SAVE R5
001240  010005                          MOV     R0,R5                   ;COPY ADDRESS IN CONTROL BLOCK
001242  012503                          MOV     (R5)+,R3                ;GET ADDRESS OF MESSAGE PREFIX
001244                                  CALL    20$                     ;DO A LITTLE COMMON CODE
001250  005715                          TST     (R5)                    ;CREATE A SECOND MESSAGE?
001252  001402                          BEQ     10$                     ;BR IF NO
001254                                  CALL    20$                     ;A LITTLE MORE COMMON CODE
001260  012605                  10$:    MOV     (SP)+,R5                ;RECOVER R5
001262  012604                          MOV     (SP)+,R4                ;RECOVER R4
001264                                  RETURN


001266  012704  000034'          20$:    MOV     #KMCEMS,R4              ;PLACE MESSAGE HERE
001272                                  CALL    @(R5)+                  ;FORMAT FIRST MESSAGE
001274  012700  000024'                  MOV     #KMCEBF,R0              ;GET THE FULL BUFFER ADDRESS
001300  010401                          MOV     R4,R1                   ;COPY ADDRESS OF END OF MESSAGE
001302  160001                          SUB     R0,R1                   ;GET LENGTH OF MESSAGE
001304                                  CALL    $TTOUT                  ;PRINT THE MESSAGE
001310                                  RETURN
```

```
                                .SBTTL  MACRO DEFINITIONS
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  ASL$,EMSG$R,EMSG$,SLTDF$,XPDDB$,CTRDF$,BIAS,ASR$,RETC,ECDDB$
                                .MCALL  DHBDF$,SAVRG,RESRG,CACDF$,ADJDF$

000000                          ADJDF$                  ; DEFINE ADJACENCY DATA BASE SYMBOLS
000000                          CTRDF$                  ; DEFINE ECL COUNTERS OFFSETS AND SYMBOLS
000000                          ECDDB$                  ; DEFINE ECL DDB OFFSETS AND SYMBOLS
000000                          XPDDB$                  ; DEFINE XPT'S DDB OFFSETS AND SYMBOLS
000000                          SLTDF$                  ; DEFINE SLT OFFSETS AND SYMBOLS
000000                          DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS
000000                          CACDF$                  ; DEFINE CACHE OFFSETS

                                .IF DF  R$$MPL

                                .MCALL  CTBDF$

                                CTBDF$                  ; DEFINE THE CTB OFFSETS

                                .ENDC
                        ;
                        ; LOCAL MACRO DEFINITIONS
                        ;
                                .MACRO  SAVMAP
                                MOV     @KSAR5,-(SP)    ; SAVE APR 5
                                .ENDM

                                .MACRO  RESMAP
                                MOV     (SP)+,@KSAR5    ; RESTORE APR5
                                .ENDM

                                .MACRO  BIAS5 REG
                                BIC     #160000,REG
                                BIS     #120000,REG
                                .ENDM

                                .MACRO  MAP BIAS
                                MOV     BIAS,@KSAR5
                                .ENDM
```

```
                                        .SBTTL  $ALMHC - ALLOCATE MINHOPS/MINCOST MATRIX

                                ;+
                                ; $ALMHC - ALLOCATE MIN HOPS/MIN COST MATRIX FROM POOL..
                                ;
                                ; INPUTS:
                                ;       $PDVA - XPT'S PDV ADDRESS
                                ;       $NA - MAXIMUM NUMBER OF AREAS
                                ;       $NN - MAXIMUM NODE ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       N$MHC1   (XPT'S DDB) - NUMBER OF WORDS IN MATRIX
                                ;       N$MHC1+2 (XPT'S DDB) - APR BIAS OF MATRIX
                                ;       N$MHC1+4 (XPT'S DDB) - APR6 VIRTUAL ADDRESS OF MATRIX
                                ;       N$MHC2   (XPT'S DDB) - NUMBER OF WORDS IN MATRIX
                                ;       N$MHC2+2 (XPT'S DDB) - APR BIAS OF MATRIX
                                ;       N$MHC2+4 (XPT'S DDB) - APR6 VIRTUAL ADDRESS OF MATRIX
                                ;
                                ;-
001022                          $ALMHC::CALL    $SAVAL                  ; SAVE ALL REGISTERS
001026  017700  000000G                 MOV     @DECPT,R0               ; STORE DECNET HOME BLOCK ADDRESS
001032  016001  000060                  MOV     D$NN(R0),R1             ; GET MAXIMUM NUMBER OF NODES
001036  001464                          BEQ     40$                     ; IF ENDNODE - BRANCH
001040  016005  000062                  MOV     D$NA(R0),R5             ; MULTIPLE AREAS ?
001044  001401                          BEQ     10$                     ; IF NO - BRANCH
001046  005201                          INC     R1                      ; INCLUDE EXTRA AREA ENTRY
001050  005201                  10$:    INC     R1                      ; INCLUDE EXTRA NODE ENTRY
001052  060501                          ADD     R5,R1                   ; ADD NUMBER OF AREAS
001054  006301                          ASL     R1                      ; CONVERT TO BYTE COUNT (1 WORD/NODE)
001056  062701  000077                  ADD     #77,R1                  ; ROUND TO CLOCK BYTE BOUNDARY
001062  042701  000077                  BIC     #77,R1                  ; ...
001066                                  ASR$    6,R1                    ; CONVERT TO BLOCK COUNT
001102                                  CALL    $ALN18                  ; ALLOCATE COUNTERS FROM NON-UMR SPACE
001106  103441                          BCS     101$                    ; BR IF ERROR
001110  016705  000000G                 MOV     $PDVA,R5                ; GET XPT'S PDV ADDRESS
001114  016505  000000G                 MOV     Z.DAT(R5),R5            ; GET XPT'S DDB ADDRESS
001120  017702  000000G                 MOV     @DECPT,R2               ; STORE DECNET HOME BLOCK OFFSET
001124  016201  000060                  MOV     D$NN(R2),R1             ; GET NUMBER WORDS IN NODE HOPS/COST
001130  005201                          INC     R1                      ; INCLUDE EXTRA NODE ENTRY
001132  010165  000036                  MOV     R1,N$MHC1(R5)           ; STORE NUMBER OF WORDS IN MATRIX
001136  006301                          ASL     R1                      ; CONVERT TO BYTES
001140  010065  000040                  MOV     R0,N$MHC1+2(R5)         ; STORE APR BIAS FOR MIN HOPS/MIN COST
001144  005004                          CLR     R4                      ; POINT TO START OF MATRIX
001146                                  BIAS    R4                      ; ... (XPT NEEDS APR6 BIAS)
001156  010465  000042                  MOV     R4,N$MHC1+4(R5)         ; STORE START OF MATRIX
001162  060104                          ADD     R1,R4                   ; POINT TO START OF AREA HOPS/COST
001164  016201  000062                  MOV     D$NA(R2),R1             ; GET NUMBER OF ENTRIES IN AREA HOPS/COST
001170  001407                          BEQ     40$                     ; IF NONE - BRANCH
001172  005201                          INC     R1                      ; INCLUDE EXTRA AREA ENTRY
001174  010165  000044                  MOV     R1,N$MHC2(R5)           ; STORE NUMBER OF WORDS IN AREA MATRIX
001200  010065  000046                  MOV     R0,N$MHC2+2(R5)         ; STORE APR BIAS FOR AREA MATRIX
001204  010465  000050                  MOV     R4,N$MHC2+4(R5)         ; STORE VIRTUAL ADDR OF START OF MATRIX
001210                          40$:    RETURN

001212                          101$:   EMSG$R  69                      ; ELSE ALLOCATION FAILURE
```

```
                        .TITLE  NTLROO - NTL ROOT SEGMENT
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - MAIN CONTROL LOOP
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                        .SBTTL  $VCTLB - SET UP LIBRARY VECTOR TABLE
                ;+
                ;
                ; $VCTLB - SET UP LIBRARY VECTOR TABLE
                ;
                ; INPUTS:
                ;       $LBLN - NUMBER OF SYMBOLS
                ;       $LBVT - ADDRESS OF TABLE
                ;       $LBBLK - BIAS OF LIBRARY
                ;
                ; OUTPUTS:
                ;       VECTOR TABLE TRANSLATED
                ;
                ;-
                        .IF DF  R$$MPL

                $VCTLB::MOV     $LBVL,R2        ; COPY NUMBER OF SYMBOLS VECTORED
                        BEQ     10$             ; IF PROCESS NOT VECTORED - BRANCH
                        EMT     376             ; SWSTK$ 10$ WITHOUT RSXMC
                        .WORD   10$             ;
                        MOV     @KSAR6,-(SP)    ; SAVE OUR APR6 MAPPING
                        MOV     MPPRO,R0        ; GET ADDRESS OF PROCESS MAPPING ROUTINE
                        MOV     KSAR6,R5        ; GET KISAR6 ADDRESS
                        MOV     $LBBLK,@KSAR6   ; MAP TO PROCESS
                        MOV     $LBVT,R3        ; COPY TABLE ADDRESS OF PROCESS

                        MOV     @#112,R1        ; COPY BIAS OF EXEC ROUTINE WHICH
                                                ; CALLS THE VECTOR TRANSLATE ROUTINE
                        MOV     (R1),R1         ; GETTING TO ACTUAL BIAS
                        MOV     (R1),-(SP)      ; PUT BIAS ON STACK FOR $MPPRO
                        MOV     #120000,-(SP)   ; MAP THRU APR5
                        CALL    (R0)            ; CALL DIRECTIVE COMMON ROUTINE THE LONG WAY
                        MOV     (SP)+,(R5)      ; RESTORE OUR APR6 MAPPING
                        CLR     $LBVL           ; INIT VECTOR TABLE FLAG

                10$:    RETURN

                        .ENDC
```

SYMBOL	VALUE		REFERENCES						
Z.PCB	= ******	GX	8-507	*8-556	*11-685	*11-702	12-750	*14-824	
$ALECL	= ******	GX	8-538						
$ALMHC	= ******	GX	8-533						
$ALOCB	= ******	GX	6-211						
$ALOC1	= ******	GX	6-210						
$ALTLC	= ******	GX	11-710						
$BIAS	000066	RG	#6-294	*12-752					
$CCBAF	= ******	GX	5-151						
$CCBAL	= ******	GX	5-155						
$CCBCT	= ******	GX	5-148						
$CCBLH	= ******	GX	6-188						
$CCBNM	= ******	GX	5-149						
$CCBS?	= ******	GX	6-189						
$CCEX	= ******	GX	6-407	13-797					
$CEAVL	= ******	GX	6-198						
$CEPWR	= ******	GX	5-145						
$CLIN	= ******	GX	6-409						
$CLKHD	= ******	GX	5-109						
$CMFRK	= ******	GX	5-127						
$CPRO	= ******	GX	6-408						
$CRAVL	= ******	GX	5-111						
$CTBAD	000020	RG	#6-268						
$CTIM	000106	RG	#6-301						
$CTSYM	= ******	GX	11-703						
$CXOPT	= ******	GX	6-192						
$DDMPV	000142	RG	#6-318	11-688					
$DEACB	= ******	GX	5-116						
$DEAC1	= ******	GX	5-120						
$DECPT	= ******	GX	6-168						
$DEUMR	= ******	GX	5-90						
$DEVHD	= ******	GX	5-115						
$DLCPV	000140	RG	#6-317	11-670					
$DSINI	= ******	GX	8-551						
$EXRQN	= ******	GX	5-91						
$EXSIZ	= ******	GX	5-110						
$FEATR	000026	RG	#6-271						
$FILHD	= ******	GX	6-169						
$FLAGS	001232	RG	#6-381	*8-516	8-521	*11-679	*11-696	*12-737	*12-738	*14-823
$FMASK	= ******	GX	5-108						
$FUNC	000012	RG	#6-227	7-475	7-481	*7-485			
$HEADR	= ******	GX	6-212						
$ICBI	= 000016	G	#6-232	6-243					
$ICBLN	000016	RG	#6-243						
$INICO	= ******	GX	8-530						
$LDBAF	= ******	GX	5-153						
$LDPWF	= ******	GX	5-121						
$LGDDB	= ******	GX	6-183						
$LGPDV	= ******	GX	6-182						
$LGSTT	= ******	GX	6-170						
$LLCAL	= ******	GX	6-180						
$LLCTA	= ******	GX	5-157						
$LTMFC	= ******	GX	5-150						

```
    187 001444      000                     FMT7:   .ASCIZ  ""
    188 001445      103     150     141     FMT7A:  .ASCIZ  "Channel * still active on process *"
    189 001511      114     151     156     FMT7B:  .ASCIZ  "Line * "
    190                                             .EVEN
    191                                             .DSABL  LC
    192                                             .LIST   BEX
    193
    194 000000                                      .PSECT
```

```
                                        .SBTTL  CPRO - CHECK CLEAR PROCESS REQUEST BLOCK

                                ;+
                                ; CPRO - CHECK "CLEAR PROCESS" REQUEST BLOCK
                                ;
                                ; THIS ROUTINE IS CALLED TO CHECK THAT A CLEAR PROCESS REQUEST IS VALID. IF
                                ; THE PROCESS IS AN LLC, ALL ITS CHANNELS MUST BE INACTIVE AND, IF IT IS ALSO
                                ; AN RSX11M PSEUDO-DEVICE, THE DEVICE MUST BE DISMOUNTED. IF THE PROCESS IS
                                ; EITHER A DLC OR A DDM, ALL LINES WHICH INCLUDE THE PROCESS MUST BE NOT READY.
                                ;
                                ; INPUTS:
                                ;       $PDVA=PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R4 CLEARED IF PROCESS IS NON-RESIDENT
                                ;
                                ;-

001306  005000                  CPRO:   CLR     R0              ; USED AS ERROR FLAG
001310  016701  000000G                 MOV     $PDVA,R1        ; GET PDV ADDRESS

001314  005761  000000G                 TST     Z.PCB(R1)       ; IS PROCESS CORE RESIDENT?
001320  001003                          BNE     10$             ; IF YES - BRANCH
001322  005004                          CLR     R4              ; INDICATE NO WORK TO DO
001324  000167  000522                  JMP     150$

001330  016702  000000G          10$:    MOV     $NAME,R2        ; GET PROCESS NAME

                                        .IF DF R$$MPL
                                        CMP     #^RDLX,R2       ; IS IT "DLX"?
                                        BEQ     12$             ; IF EQ, YES .. ERROR
                                        .ENDC
001334  022702  015355                  CMP     #^RDLM,R2       ; IS IT "DLM"?
001340  001403                          BEQ     12$             ; IF EQ, YES .. ERROR
001342  022702  004640                  CMP     #^RAUX,R2       ; IS IT "AUX"?
001346  001003                          BNE     14$             ; NO ..

001350                           12$:   EMSG$R  ZR              ; PROCESS MAY NOT BE UNLOADED

001356  032767  000000G 000000G 14$:    BIT     #FL.LLC,$FLAGS  ; IS THIS AN LLC?
001364  001573                          BEQ     100$            ; IF EQ, NO
001366  005002                          CLR     R2              ; GET CHANNEL COUNT
001370  156102  000000G                 BISB    Z.LLN(R1),R2    ;   ...
001374  001473                          BEQ     40$             ; IF ZERO, NONE TO CHECK
001376  006302                          ASL     R2              ; CONVERT TO A WORD INDEX
001400  062701  000000G                 ADD     #Z.MAP,R1       ; POINT AT MAPPING TABLE
001404  060102                          ADD     R1,R2           ; GIVING END OF MAPPING TABLE ADDRESS
001406  022721  140000                  CMP     #140000,(R1)+   ; IS CHANNEL 0 ASSIGNED?
001412  001462                          BEQ     30$             ; IF EQ, NO
001414  005741                          TST     -(R1)           ; RESET CHANNEL POINTER
001416  022721  100000          20$:    CMP     #100000,(R1)+   ; IS CHANNEL 1-N ASSIGNED?
001422  001456                          BEQ     30$             ; IF EQ, NO
001424  017746  000000G                 MOV     @SLTMA,-(SP)    ; SAVE START OF SLT MAPPING TABLE

001430  012746  001472'                 MOV     #25$,-(SP)      ; TO ALLOW CO-ROUTINE CLEANUP
```

```
                                        .SBTTL  $RERR - PRINT ERROR MESSAGE

                                ;+
                                ;
                                ; $RERR - PRINT ERROR MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;       R0,R1=DESTROYED
                                ;
                                ;-

003332                          $RERR::
003332                                  CALL    $SAVAL          ; SAVE ALL REGISTERS
003336  010005                          MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
003340  012704  000170'                 MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
003344  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
003346                                  CALL    @(R5)+          ; DO THE FORMATTING
003350  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
003352  001403                          BEQ     20$             ; IF EQ, YES
003354  112324                  10$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
003356  001376                          BNE     10$             ;       ...
003360  005304                          DEC     R4              ; SKIP THE ZERO BYTE
003362  012503                  20$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
003364  112524                  30$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
003366  001376                          BNE     30$             ;       ...
003370  005304                          DEC     R4              ; SKIP THE ZERO BYTE
003372  012700  000160'                 MOV     #ERRMSG,R0      ; GET ERROR MESSAGE ADDRESS
003376  010401                          MOV     R4,R1           ; COPY END ADDRESS
003400  160001                          SUB     R0,R1           ; GIVING LENGTH
003402                                  CALL    $TIOUT          ; PRINT IT
003406  005703                          TST     R3              ; SHOULD SECOND LINE BE PRINTED?
003410  001401                          BEQ     40$             ; IF EQ, NO
003412                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
003414  000261                  40$:    SEC                     ; RETURN WITH C-BIT SET
003416                                  RETURN
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-59	15-543								
CALL	8-231	11-333	13-406	14-443	14-467	15-534	15-567	17-626	18-713	18-718
	18-726	18-776	20-833	20-837	21-942	21-951	21-1001	22-1027	22-1033	22-1042
	26-1205	26-1209	26-1222	26-1225	28-1265	28-1278	28-1287	29-1310	30-1339	30-1351
	30-1362	30-1368								
EMSG$	#5-59	18-713	18-718	18-776	21-942	21-1001				
EMSG$R	#5-59	8-241	9-284	10-310	11-345	11-346	14-487	14-488	14-489	15-586
	15-587	15-588	15-589	15-590	18-672	18-754	18-757	19-809	21-977	21-980
	22-1055	25-1187								
NTLDF$	#5-59	5-61								
NTLER$	#5-59	7-145	7-146	7-147	7-148	7-149	7-150	7-151	7-152	7-154
	7-158	7-159	7-160	7-161	7-167	7-168	7-169	7-172	7-175	7-176
	7-177	7-178								
RETURN	8-236	9-286	10-305	11-340	12-389	13-419	14-482	15-581	16-609	17-631
	18-697	18-751	18-779	19-804	20-844	20-846	21-938	21-973	21-1008	22-1051
	23-1083	24-1123	25-1181	26-1227	27-1244	28-1289	29-1318	30-1370		
SLTDF$	#5-59	5-62								
SOB	9-277	18-777	21-1002	24-1112						
SWSTK$	18-726	20-833	21-951							

NTLSUB - COMMON SUBROUTINES MACRO V05.03b Saturday 29-Jun-85 00:33 Page 13-1
Symbol table

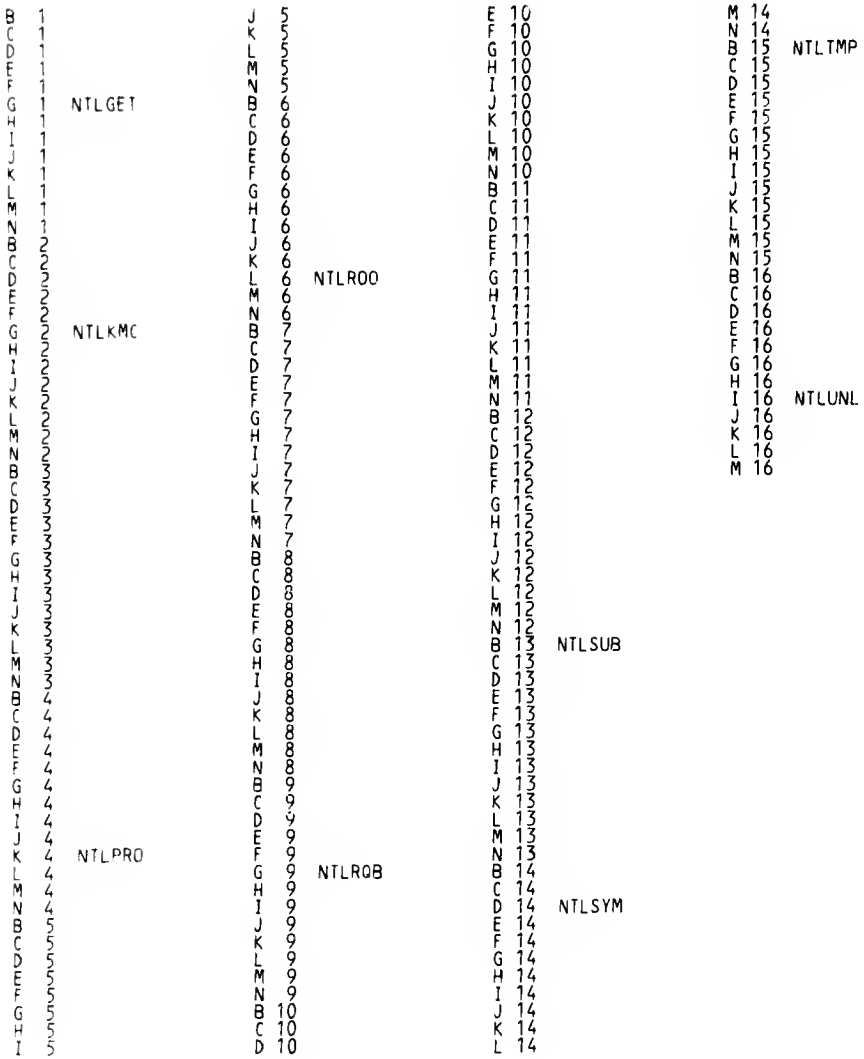
```
A$$CHK= 000000          G$$TPP= 000000          M$$MGE= 000000          Q.IOPR= 000007          $CFUIC  000052RG    002
A$$CPS= 000000          G$$TSS= 000000          M$$NET= 000000          Q.IOSB= 000010          $CLEVL= ****** GX
A$$PRI= 000000          G$$TTK= 000000          M$$OVR= 000000          R$$DER= 000000          $CLIOS= ****** GX
A$$TRP= 000000          G$$WRD= 000000          NAMES   000125R  002  R$$K11= 000001          $CUR.N  000216RG
C$$CKP= 000000          IO.WBT= ****** GX       N$$ACC= 000001          R$$SND= 000000          $CXUIC  000072RG    002
C$$ORE= 000400          I$$RAR= 000000          N$$BUF= 000001          R$$11M= 000000          $C5TA = ****** GX
C$$RSH= 177564          I$$RDN= 000000          N$$LDV= 000001          SLFNP   000322R         $DFN    000000RG
DFEXT   000121R  002    K$$CNT= 177546          N$$MCP= 000001          S$$WRG= 000000          $DFNAM  000112RG    002
D$$BUG= 177514          K$$CSR= 177546          N$$MLL= 000001          S$$YSZ= 007600          $DFN2   000002RG
D$$ISK= 000000          K$$LDC= 000000          N$$MOV= 000010          TIOUT   000000R  002  $DFUIC  000032RG    002
D$$L11= 000001          K$$TPS= 000074          N$$NCT= 000001          T$$KMG= 000000          $FEATR= ****** GX
D$$YNC= 000000          LDPWF = ****** GX       N$$PEM= 000001          T$$MJN= 000000          $FNOUT  000270RG
D$$YNM= 000000          LD$LP = 000000          P$$P45= 000000          UC.PWF= ****** GX       $NLCDD  000352RG
D.DSP = ****** GX       L$$ASG= 000000          P$$WRD= 000000          U.CTL = ****** GX       $PRV.N  000224RG
D.UCBL= ****** GX       L$$DRV= 000000          Q$$OPT= 000010          U.CW3 = ****** GX       $REP.C  000160RG
E$$XPR= 000000          L$$P11= 000001          Q.IOAE= 000012          U.SCB = ****** GX       $REP.N  000230RG
FILE    000243R  002    L$$11R= 000000          Q.IOEF= 000006          V$$CTR= 001000          $REP.P  000164RG
FNERR   000233R  002    M$$CRB= 000124          Q.IOFN= 000002          X$$DBT= 000000          $TIOUT  000302RG
FNPTR   000030R  002    M$$CRX= 000000          Q.IOLU= 000004          $AZFS   000042RG        $$$ARG= 000003
F$$LVL= 000001          M$$FCS= 000000          Q.IOPL= 000014          $CBDSG= ****** GX       $$$OST= 000014


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000440    001   (RW,I,LCL,REL,CON)
DATA    000300    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9768 Words  ( 39 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:11.57
SY:NTLSUB.V2,[132,134]NTLSUB/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLSUB
```

NTLSYM CREATED BY MACRO ON 29-JUN-85 AT 00:34 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES
PDVDF$	#5-57 5-64

```
                                ;+
                                ; $VFDLC - VERIFY DLC ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ERROR UNLESS DLC BEING LOADED
                                ;-
000542                          $VFDLC::
000542  032767  000000G 000000G         BIT     #FL.DLC,$FLAGS  ; IS THIS A DLC?
000550  001003                         BNE     10$             ; IF NE, YES
000552                                  EMSG$   79              ; DLC OPERATOR IN DDM/LLC TEMPLATE
000560                          10$:    RETURN
```


NTLGET
NTLKMC
NTLPRO
NTLROO
NTLRQB
NTLSUB
NTLSYM
NTLTMP
NTLUNL

```
                                .SBTTL  LOCAL DATA

                        ;***
                        ; LOCAL DATA
                        ;***

                                .PSECT  DATA,D
000000                  K6TMP:  .BLKW   1               ; $K6TAB TEMPORARY LOCATION
000002                  FMKTMP: .BLKW   1               ; $FMASK TEMPORARY LOCATION
000004                  URMTMP: .BLKW   1               ; $URMTB TEMPORARY LOCATION
000006                  ICBSZ:  .BLKW   1               ; SIZE OF ICB BLOCK

                                .IF DF  R$$MPL
                        BIAS:   .BLKW   1               ; LOCAL COPY IF DDM PROCESS BIAS
                        KRBA:   .BLKW   1               ; KRB ADDRESS
                                .ENDC   ; DF R$$MPL

                        ;
                        ; UNLOAD BLOCK BIAS AND ADDRESS
                        ;
000010                  UBIAS:  .BLKW   1               ; START OF BLOCK ADDRESS UNMAPPED
000012                  UBA:    .BLKW   1               ; UNLOAD BLOCK ADDRESS

                        ;
                        ; COPY OF UNLOAD BLOCK
                        ;
000014                  BLOCK:  .BLKW   32.

                        ;
                        ; POINTER INTO UNLOAD BLOCK COPY
                        ;
000114                  UPTR:   .BLKW   1

                        ;
                        ; BIAS OF BLOCK TO DEALLOCATE (IN "KMCRL")
                        ;
000116                  KMCB:   .IIF    DF,M$$MGE,      .BLKW   1

                        ;
                        ; LOOP COUNTER
                        ;
000120                  COUNT:  .BLKW   1

                        ;
                        ; "CLEAR LINE" COUNTERS
                        ;
000122                  DLCAC:: .BLKW   1               ; DLC ACTIVITY COUNT
000124                  DDMAC:: .BLKW   1               ; DDM ACTIVITY COUNT

                        ;
                        ; NONSENSE INTERRUPT TABLE (RSX11M ONLY)
                        ;
000126  000000G         NSITAB: .WORD   NS0
000130  000000G                 .WORD   NS1
000132  000000G                 .WORD   NS2
000134  000000G                 .WORD   NS3
000136  000000G                 .WORD   NS4
```

```
662                                     ;
663 001406                          101$:   CRASH                   ; KRB ADDRESS IS ZERO !
664 001410                          111$:   CRASH                   ; DLC LINE TABLE ADDRESS IS ZERO !
```

```
                                        MOV     @NTLPT,R2       ; GET THE NTL HOME BLOCK ADDRESS
                                        BEQ     83$             ; IF EQ, THERE IS NONE
                                        MOV     .CXALL(R2),R0   ; GET THE CETAB ALLOCATION ADDRESS
                                        MOV     .CXLEN(R2),R1   ; GET THE CETAB ALLOCATION LENGTH

                                        .ENDC

002734  005700                 83$:    TST     R0              ; IS THERE A CETAB ALLOCATION ?
002736  001402                         BEQ     84$             ; IF EQ, NO
002740                                  CALL    $DEA16          ; DE-ALLOCATE THE CETAB ALLOCATION
                                        .IFTF   ; DF R$$11M
002744                          84$:
                                ;
                                ; CLEAN-UP POOL TABLES
                                ;
002744                                  CALL    XOBJTB          ; DE-ALLOCATE ALL OBJECTS
002750                                  CALL    XALITB          ; DE-ALLOCATE ALL ALIASES
002754                                  CALL    XREMTB          ; DE-ALLOCATE ALL REMOTES
002760                                  CALL    XFLTTB          ; DE-ALLOCATE ALL FILTER CONTROL BLOCKS

002764                                  CALL    $XPSI           ; DE-ALLOCATE PSI DATA BASES FROM EXTENDED POOL
002770  005077  000000G                 CLR     @CXOPT          ; NO COMM EXEC OPTIONS SET NOW

                                        .IFT    ; DF R$$11M

002774  016700  000000G                 MOV     .PCB,R0         ; GET NTPOOL PCB ADDRESS
003(  001417                            BEQ     90$             ; IF NONE, EXIT
0030(2                                  CALL    $XPAR           ; RELEASE PARTITION
003006  10341b                          BCS     101$            ; IF CS, VERY BAD

                                        .IFF    ; DF R$$11M

                                        MOV     .REG,R0         ; GET REGION GCD ADDRESS
                                        BEQ     90$             ; BR IF NONE
                                        CALL    $ULOCK          ; UNLOCK THE REGION

                                        .IFT    ; DF R$$11M
                                        .IF DF  M$$MGE
                                        .IF DF  R$$MPL
                                        MOVB    @NCPU,R4        ; GET THE NUMBER OF PROCESSORS
                                        BEQ     85$             ; IF EQ, NOT A MULTIPROCESSOR SYSTEM
                                        CALL    XUMR            ; RELEASE THE UMR'S
                                        BR      90$             ; AND LEAVE
                                85$:
                                        .ENDC

003010  016700  000000G                 MOV     .UMRBA,R0       ; GET UMR BLOCK ADDRESS
003014  001411                          BEQ     90$             ; IF NONE, EXIT
003016                                  SWSTK$  90$             ;; ENTER KERNEL MODE
003022  010002                          MOV     R0,R2           ;; COPY UMR BLOCK ADDRESS
003024                                  CALL    @DEUMR          ;; DE-ASSIGN UMR'S
003030  012701  000000G                 MOV     #M.LGTH,R1      ;; GET UMR BLOCK LENGTH
003034                                  CALLR   $DECEX          ;; DE-ALLOCATE UMR BLOCK
                                        .ENDC

                                        .IFTF   ; DF R$$11M
```

```
                                        SWSTK$  300$             ;; ENTER KERNEL MODE
                                                .ENDC
004142                                  CALL    $MAPX            ;; MAP TO THE UNLOAD BLOCK
                                        .IFF
                                        MOV     UBIAS,R2         ; GET UNLOAD BLOCK ADDRESS
                                        .ENDC

004146  012700  000014'                 MOV     #BLOCK,R0        ;; REWRITE UNLOAD BLOCK FROM LOCAL COPY
004152  012701  000040                  MOV     #32.,R1          ;;      ...
004156  012022          290$:           MOV     (R0)+,(R2)+      ;;      ...
004160                                  SOB     R1,290$          ;;      ...

004164                  300$:           .IF DF R$$11D ! I$$AS
                                        CALL    @(SP)+           ;; RESTORE PREVIOUS MAPPING
                                        .ENDC

004164                                  RETURN                   ;; DONE (FINALLY!)
```



```
NTLUNL

SYMBOL (

SYMBOL

EV$RSC
EV$RUL
EV$SNA
EV$SNF
EV$SPE
EV$XCE
EV$XDI
EV$XGW
EV$XMX
EV$XRS
EV$XSC
EV$X2S
EV.CCB
EV.CIR
EV.LCB
EV.LIN
EV.MAP
EV.MOD
EV.NOD
EV.PRT
EXRQN
E$DATA
E$EVTS
E$LCN
E$LIN
E$MOD
E$NBR
E$NBS
E$NCR
E$NCS
E$NIC
E$NLEN
E$NLLA
E$NLNK
E$NML
E$NMR
E$NMS
E$NNOD
E$NOD
E$NRT
E$NRTP
E$NSEG
E$NTIM
E$NUSE
E$PORT
E$PRM
E$STAT
E$STRT
E$TCB
E.CTL
E.DATA
E.EVT
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES	
EV$RSC	= 000501		#5-66	
EV$RUL	= 000405		#5-66	
EV$SNA	= 035000		#5-66	
EV$SNF	= 000516		#5-66	
EV$SPE	= 035001		#5-66	
EV$XCE	= 034110		#5-66	
EV$XDI	= 013600		#5-66	
EV$XGW	= 034111		#5-66	
EV$XMX	= 000514		#5-66	
EV$XRS	= 000512		#5-66	
EV$XSC	= 000513		#5-66	
EV$X2S	= 013500		#5-66	
EV.CCB	= 000001		#5-66	
EV.CIR	= 000020		#5-66	
EV.LCB	= 000100		#5-66	
EV.LIN	= 000004		#5-66	
EV.MAP	= 000002		#5-66	
EV.MOD	= 000040		#5-66	
EV.NOD	= 000010		#5-66	
EV.PRT	= 000200		#5-66	
EXRQN	= ******	GX	8-286	
E$DATA	000020		#5-66	
E$EVTS	000000		#5-66	
E$LCN	000016		#5-66	
E$LIN	000000		#5-66	
E$MOD	000012		#5-66	
E$NBR	000014		#5-64	
E$NBS	000020		#5-64	
E$NCR	000034		#5-64	
E$NCS	000036		#5-64	
E$NIC	000044		#5-64	
E$NLEN	000050		#5-64	10-380
E$NLLA	000012		#5-64	
E$NLNK	000000		#5-64	
E$NML	000040		#5-64	
E$NMR	000024		#5-64	
E$NMS	000030		#5-64	
E$NNOD	000002		#5-64	
E$NOD	000010		#5-66	
E$NRT	000042		#5-64	
E$NRTP	000005		#5-64	
E$NSEG	000010		#5-64	
E$NTIM	000046		#5-64	
E$NUSE	000004		#5-64	
E$PORT	000014		#5-66	
E$PRM	000002		#5-66	
E$STAT	000006		#5-66	8-295
E$STRT	000006		#5-64	
E$TCB	000004		#5-66	8-282 *8-284
E.CTL	000020		#5-66	
E.DATA	000046		#5-66	
E.EVT	000002		#5-66	

NTL11M -

112
113
114
115
116
117
118
119
120

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 4

NTL11M -

```
                                ; EXECUTIVE REFERENCES
                                ;***
                                        .GLOBL  $ALOCB,$ALOC1,$CRAVL,$DEACB,$DEAC1,$DEVHD,$EXSIZ,$FMASK,$PARHD
                                        .GLOBL  $NS0,$NS1,$NS2,$NS3,$NS4,$NS5,$NS6,$NS7,$TKTCB,$CLKHD,.TTO
                                        .GLOBL  P.BLKS,P.LGTH,P.REL,P.SWSZ,P.SIZE
                                        .IIF DF M$$MGE, .GLOBL $ASUMR,$DEUMR,P.ATT,P.PRO


000001                                  .END
```



```
POLAL -
$XALOC -
```

```
                                        MOV     R4,R0           ; GET ADDRESS OF BLOCK ALLOCATED
                                        MOV     (SP)+,R4        ; RESTORE
                                        MOV     (SP)+,R3        ;  REGISTERS

                                        .ENDC   ; DF R$$11M

000040  000427                          BR      30$             ; AND RETURN

000042  010046                  15$:    MOV     R0,-(SP)        ; SET UP ADDRESS FOR $CEACX
000044                                  CALL    $CEACX          ; MAP EXTENDED BLOCK
000050  012600                          MOV     (SP)+,R0        ; RETRIEVE MAPPED ADDRESS
000052  026001  000002                  CMP     2(R0),R1        ; BLOCK BIG ENOUGH?
000056  103763                          BLO     10$             ; IF LO NO
000060  001407                          BEQ     20$             ; IF EQ BLOCK IS EXACT SIZE
000062  160160  000002                  SUB     R1,2(R0)        ; CALCULATE SIZE REMAINING
000066  016046  000002                  MOV     2(R0),-(SP)     ; SAVE SIZE OF REMAINING BLOCK
000072  011200                          MOV     (R2),R0         ; RETRIEVE UNMAPPED BLOCK ADDRESS
000074  062600                          ADD     (SP)+,R0        ; CALC ADDRESS OF BLOCK RETURNED
000076  000410                          BR      30$             ; AND RETURN

000100  011246                  20$:    MOV     (R2),-(SP)      ; SAVE UNMAPPED BLOCK ADDRESS
000102  011046                          MOV     (R0),-(SP)      ; SAVE NEXT BLOCK ADDRESS
000104  010246                          MOV     R2,-(SP)        ; SET UP FOR $CEACX
000106                                  CALL    $CEACX          ; MAP ADDRESS OF BLOCK
000112  012600                          MOV     (SP)+,R0        ; GET MAPPED ADDRESS
000114  012610                          MOV     (SP)+,(R0)      ; UNLINK BLOCK FROM LIST
000116  012600                          MOV     (SP)+,R0        ; AND RETURN UNMAPPED ADDRESS TO USER
000120                          30$:    RESMAP                  ; RESTORE APR 5 MAPPING
000124  016777  177652  000000G         MOV     XAVLL,@XAVL     ; UPDATE CEX COPY OF LISTHEAD
000132                                  RETURN                  ;

                                        .ENDC
```

```
                                .SBTTL  MACRO DEFINITIONS
                        ;***
                        ; MACRO CALLS
                        ;***
                                .IF DF  R$$11D!I$$AS
                                .MCALL  .INHO,.ENBO
                                .ENDC
```

LOCAL DATA

```
   192 000000                      DSKBUF: .BLKW   1
   193
   194 000000                              .PSECT
```

```
                        MOV     (SP)+,R4        ; RESTORE REGISTERS
                        MOV     (SP)+,R3        ;
                        CLC                     ; INDICATE SUCCESS
                        RETURN

101$:   MOV     #ERR36,ERROR    ; FAILED TO CREATE PROCESS REGION
        BR      105$
102$:   MOV     #ERR38,ERROR    ; FAILED TO MAP IO PROCESS REGION
105$:   SEC                     ; INDICATE FAILURE
        RETURN

        .ENDC
```

```
                        .SBTTL  $DEAC2 - DEALLOCATE FROM END OF PARTITION
                ;+
                ; *** - $DEAC2 - DEALLOCATE POOL SPACE AT END OF PROCESS PARTITION
                ;
                ; INPUTS:
                ;       R0 = ADDRESS OF THE CORE BUFFER TO BE DEALLOCATED.
                ;       R1 = SIZE OF THE CORE BUFFER TO DEALLOCATE IN BYTES.
                ;       R2 = DISPLACEMENT ADDRESS (100000 OR 120000 FOR RSX11M/M+).
                ;                                 (40000 FOR IAS)
                ;       R3 = ADDRESS OF CORE ALLOCATION LISTHEAD-2.
                ;
                ; OUTPUTS:
                ;
                ;       THE CORE BLOCK IS MERGED INTO THE FREE CORE CHAIN BY CORE
                ;       ADDRESS AND IS AGLOMERATED IF NECESSARY WITH ADJACENT BLOCKS.
                ;
                ;-
                ;
000076          $DEAC2::
000076  061301          ADD     (R3),R1         ;ROUND TO NEXT BOUNDARY
000100  042301          BIC     (R3)+,R1        ;CLEAR EXCESS
000102  001452          BEQ     60$             ;IF EQ NO BLOCK TO RELEASE
000104  010246          MOV     R2,-(SP)        ;SAVE DISPLACEMENT
000106  010302  20$:     MOV     R3,R2           ;SAVE ADDRESS OF CURRENT BLOCK
000110  011203          MOV     (R2),R3         ;GET ADDRESS OF NEXT BLOCK
000112  001405          BEQ     30$             ;IF EQ END OF CHAIN
000114  042703  160000  BIC     #160000,R3      ;CLEAR CURRENT MAPPING REG.
000120  051603          BIS     (SP),R3         ;USE OUR MAPPING REG.
000122  020003          CMP     R0,R3           ;BLOCK GO HERE?
000124  103370          BHIS    20$             ;IF HIS NO
000126  010310  30$:     MOV     R3,(R0)         ;ASSUME NO AGLOMERATION
000130  042710  160000  BIC     #160000,(R0)    ;OTHER PEOPLE USE APR6 MAPPING
                        .IF DF  R$$11M
000134  052710  140000  BIS     #140000,(R0)    ;       ...
                        .IFF    ; DF R$$11M
                        BIS     #40000,(R0)     ;USE APR 2 MAPPING
                        .ENDC
000140  010046          MOV     R0,-(SP)        ;CALCULATE ADDRESS OF NEW BLOCK
000142  060116          ADD     R1,(SP)         ;
000144  016646  000002  MOV     2(SP),-(SP)     ;;CALCULATE END OF APR ADDRESSING
000150  052716  017777  BIS     #17777,(SP)     ;;
000154  022616          CMP     (SP)+,(SP)      ;;BIGGER THAN APR ADDRESSING RANGE?
000156  103427          BLO     80$             ;;IF LO, YES
000160  020326          CMP     R3,(SP)+        ;EQUAL TO NEXT IN CHAIN?
000162  103423          BLO     70$             ;;IF LO, DEALLOCATION OVERLAPS
000164  001002          BNE     40$             ;IF NE NO
000166  012310          MOV     (R3)+,(R0)      ;MOVE LINK WORD TO BLOCK RELEASED
000170  061301          ADD     (R3),R1         ;MERGE TWO BLOCKS
000172  010246  40$:     MOV     R2,-(SP)        ;SAVE ADDRESS OF PREVIOUS BLOCK
000174  010012          MOV     R0,(R2)         ;ASSUME NO AGLOMERATION
000176  042712  160000  BIC     #160000,(R2)    ;OTHER PEOPLE USE APR6 MAPPING
```

```
                                        .SBTTL  REP5A - REPLACE PARTITION NAME
                                ;+
                                ; REP5A - REPLACE PARTITION NAME
                                ;
                                ; INPUTS:
                                ;       $PNAME=PARTITION NAME
                                ;       R3=FORMAT STRING ADDRESS
                                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       R3,R4=UPDATED
                                ;-
000154                          REP5A:  CALL    $REP.P          ; REPLACE PROCESS NAME
000160  112314                  10$:    MOVB    (R3)+,(R4)      ; LOOK FOR SECOND ASTERISK
000162  122724  000052                  CMPB    #'*,(R4)+       ;       ...
000166  001374                          BNE     10$             ;       ...
000170  005304                          DEC     R4              ; SKIP THE ASTERISK
000172  010400                          MOV     R4,R0           ; COPY BUFFER ADDRESS
000174  016701  000000G                 MOV     $PNAME,R1       ; GET FIRST HALF OF NAME
000200                                  CALL    $C5TA           ; CONVERT TO ASCII
000204  016701  000002G                 MOV     $PNAME+2,R1     ; GET SECOND HALF OF NAME
000210  001402                          BEQ     20$             ; IF ZERO, DON'T BOTHER CONVERTING
000212                                  CALL    $C5TA           ; CONVERT TO ASCII
000216  122740  000040          20$:    CMPB    #' ,-(R0)       ; SKIP TRAILING BLANKS
000222  001775                          BEQ     20$             ;       ...
000224  005200                          INC     R0              ;       ...
000226  010004                          MOV     R0,R4           ; SET NEW BUFFER ADDRESS
000230                                  RETURN
```

$ADJ - ALLOCATE ADJACENCY DATA BASE

```
000402  062701  000077             ADD     #77,R1          ;ROUND UP TO NEXT BLOCK
000406  042701  000077             BIC     #77,R1          ;
000412                             ASR$    6,R1            ;CONVERT TO BLOCKS
000426                             CALL    $ALN18          ;ALLOCATE SECOND ADJACENCY
000432  103004                     BCC     50$             ;IF SUCCESS - BRANCH
000434                             EMSG$   9D              ;ELSE - PRINT ERROR
000442                             RETURN

000444  010502              50$:   MOV     R5,R2           ;RESTORE XPT DDB ADDRESS
000446  010062  000074             MOV     R0,N$ADJ2(R2)   ;STORE SECOND ADJ. BIAS
000452  000723                     BR      20$             ;CONTINUE
000454  005062  000072     60$:    CLR     N$ADJ1(R2)      ;INIT NI CACHE SIZE
000460  005062  000074             CLR     N$ADJ2(R2)      ;INIT NI CACHE ADDRESS
000464                     70$:    RETURN
```

```
                                        ;
000162                          40$:    CALL    $CLDEQ          ; CLOSE SYMBOL TABLE FILE
000166  012636                  50$:    MOV     (SP)+,@(SP)+    ; RESTORE DEFAULT EXTENSION
000170  166666  000006  000004          SUB     6(SP),4(SP)     ; CALCULATE NUMBER OF UNDEFINEDS
000176  006266  000002                  ASR     2(SP)           ; SET C-BIT APPROPRIATELY
000202                                  RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000204  005266  000006          101$:   INC     6(SP)           ; NO BLOCKS TO CHECK
000210  000766                          BR      50$
000212  012766  000005  000006  111$:   MOV     #<<E   >EN+2>+1>,6(SP) ; OPEN FAILURE
000220  000762                          BR      50$
```

```
                                ;+
                                ; $CSR - ".CSR" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $DIFF=VALUE TO BE ADDED TO CSR ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000176                          $CSR::
000176                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000202                                  CALL    $RQ2            ; REQUEST BINARY FILE
000206  103407                          BCS     10$             ; IF CS, NO ACCESS
000210  112722  000000G                 MOVB    #.CSR,(R2)+     ; STORE FUNCTION CODE
000214  116712  000000G                 MOVB    $DIFF,(R2)      ; STORE CSR MODIFICATION VALUE
000220  062767  000002  000000G         ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
000226                          10$:    RETURN
```

```
                                        ;+
                                        ; $NOD  - ".NODE" ACTION ROUTINE
                                        ; $PROV - ".PRTOV" ACTION ROUTINE
                                        ;
                                        ; INPUTS::
                                        ;       $NOD  - NONE
                                        ;       $PROV - $VALUE = PROTOCOL OVERHEAD SIZE
                                        ;
                                        ; OUTPUTS::
                                        ;       R2 - DESTROYED
                                        ;-
                                        $NOD::  CALL    $RQ1            ; GET BUUFER ACCESS
001064  103405                          BCS     10$             ; IF NO ACCESS - BRANCH
001070  112712  000000G                 MOVB    #.NODE,(R2)     ; STORE FUNCTION CODE
001072  062767  000002  000000G         ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
001076                                  RETURN
001104
                                        $PROV::CALL     $RQ2            ; GET BUFFER ACCESS
001106  103417                          BCS     10$             ; IF NO ACCESS - BRANCH
001112  005767  000000G                 TST     $VALUE          ; PARAMETER MUST BE NON-ZERO
001114  001413                          BEQ     5$              ; IF ILLEGAL - BRANCH
001120  027767  000000G 000000G         CMP     @MAXOV,$VALUE   ; CHECK AGAINST MAX SIZE LIMIT
001122  103410                          BLO     10$             ; IF ILLEGAL - BRANCH
001130  112722  000000G                 MOVB    #.PRTOV,(R2)+   ; STORE FUNCTION CODE
001132  116712  000000G                 MOVB    $VALUE,(R2)     ; STORE VALUE
001136  005267  000000G                 INC     $SIZE           ; ADJUST DATA BASE SIZE
001142                                  RETURN
001146

001150  000261                  5$:     SEC
001152                          10$:    RETURN
```

LOCAL DATA

```
    184 000140  000000G                        .WORD   NS5
    185 000142  000000G                        .WORD   NS6
    186 000144  000000G                        .WORD   NS7
    187
    188                                 ;
    189                                 ; LOCAL COPY OF ROOT DATA NEEDED WITHIN "MAPX" CODE
    190                                 ;
    191 000146                          .NAME:  .BLKW   1               ; COPY OF $NAME
    192 000150                          .NTLPT: .BLKW   1               ; COPY OF $NTLPT FROM CETAB
    193
    194                                 ;
    195                                 ; 32. WORD BLOCK ALLOCATION SIZE
    196                                 ;
    197         000001                  BLKSIZ=1
    198
    199                                 ;
    200                                 ; PDV ADDRESS FOR PARTITION LINE TABLE DE-ALLOCATOR
    201                                 ;
    202 000152  000000                  ..PDV:  .WORD   0               ; IF ZERO, DISCARD INSTEAD OF DE-ALLOCATE
    203
    204 000000                                  .PSECT
```

```
                                        .SBTTL  VECT - RESET INTERRUPT VECTOR(S)

                                ;+
                                ; VECT - RESET INTERRUPT VECTOR(S)
                                ;
                                ; INPUTS:
                                ;       R5 - SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;-

001412                          VECT::
001412  016504  000016                  MOV     L.KRBA(R5),R4   ; GET KRB ADDRESS
001416  001002                          BNE     1$              ; IF NE, GOT IT
001420  000167  000230                  JMP     60$             ; ELSE, LEAVE

001424                          1$:
                                        .IF DF  R$$MPL
                                        MOV     R4,KRBA         ; SAVE THE KRB ADDRESS
                                        TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
                                        BEQ     2$              ; IF EQ, NO
                                        MOV     K.UBM(R4),SETAF+S.AFAF ; GET UNIBUS RUN MASK FOR DEVICE
                                        DIR$    #SETAF          ; SET AFFINITY TO DEVICE'S BUS
                                2$:
                                        .ENDC   ; DF R$$MPL

001424  116400  000001                  MOVB    K.VCT(R4),R0    ; GET VECTOR ADDRESS
001430  006300                          ASL     R0              ; CONVERT TO REAL VECTOR ADDRESS
001432  006300                          ASL     R0              ;
001434  016701  000006'         3$:     MOV     $MXVCT,R1       ; GET HIGHEST VECTOR ADDRESS IN SYSTEM
001440  005067  000006'                  CLR     ICBSZ           ; CLEAR SIZE OF ICB BLOCK
001444  016703  000012'                  MOV     UBA,R3          ; GET UNLOAD BLOCK ADDRESS
001450  001002                          BNE     4$              ; IF NE, WE FOUND IT
001452  0C   67  000200                  JMP     101$            ; ELSE, LINE TABLE NOT FOUND!
001456  042703  177700          4$:     BIC     #^C<77>,R3      ; SAVE LOW 6 BITS
001462  062703  000014'                  ADD     #BLOCK,R3       ; GIVING ADDRESS WITHIN LOCAL BLOCK
001466  005723                          TST     (R3)+           ; SKIP PDV INDEX
001470  012302                          MOV     (R3)+,R2        ; GET LINE TABLE STARTING ADDRESS
001472  011303                          MOV     (R3),R3         ; AND LENGTH
001474  042703  000003                  BIC     #3,R3           ; DISCARD STATUS BITS
001500  060203                          ADD     R2,R3           ; GIVING END OF LINE TABLE ADDRESS

001502                                  SWSTK$  50$             ;; ENTER KERNEL MODE

                                        .IF DF  R$$MPL

                                        MOV     $BIAS,BIAS      ;; GET LOCAL COPY OF PROCESS BIAS
                                        MOV     FMASK,FMKTMP    ;; GET $FMASK ADDRESS
                                        ADD     #2,FMKTMP       ;; GET $FMASK+2 ADDRESS
                                        BIT     #F2.DAS,@FMKTMP ;; ; DATA SPACE ENABLED ?
                                        BEQ     10$             ;; IF EQ, NO
                                        MOV     @KSAR5,-(SP)    ;; SAVE CURRENT MAPPING
                                        MOVB    @NCPU,R5        ;; GET THE NUMBER OF PROCESSORS
                                        BEQ     6$              ;; IF EQ, NOT A MULTIPROCESSOR SYSTEM
                                        ASL     R5              ;; CONVERT TO A WORD INDEX
```

```
1338 003040  000261                90$:    SEC                     ; RETURN C-BIT SET
1339 003042                                RETURN
1340
1341                                       .IFT    ; DF R$$11M
1342
1343                               ;
1344                               ; ERROR CONDITION
1345                               ;
1346 003044                        101$:   CRASH                   ; SOMEBODY UNLINKED THE NTPOOL PCB!
1347
1348                                       .ENDC   ; DF R$$11M
```

```
                                        .SBTTL  .DEA16 - END OF PARTITION DE-ALLOCATION ROUTINE

                                ;+
                                ; .DEA16 - END OF PARTITION DE-ALLOCATION ROUTINE
                                ;
                                ; THIS ROUTINE IS CALLED IN PLACE OF "$DEA16", TO ALLOW BOTH NORMAL AND
                                ; PARTITION ALLOCATIONS TO BE RELEASED. IF THE ALLOCATION CAME FROM THE
                                ; PROCESS' PARTITION SPACE, THE LOCATION "..PDV" WILL CONTAIN THE PDV ADDRESS
                                ; IF THE ALLOCATION IS TO BE RELEASED OR ZERO IF THE ALLOCATION IS TO BE
                                ; DISCARDED (I.E., THE PROCESS IS ABOUT TO BE RELEASED FROM ITS PARTITION SO
                                ; FURTHER HOUSEKEEPING OF THE FREE SPACE IS NOT NEEDED).
                                ;
                                ; INPUTS:
                                ;       R0 = ALLOCATION ADDRESS
                                ;       R1 = ALLOCATION SIZE
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2 = DESTROYED
                                ;       R3,R4,R5 = SAVED
                                ;-
004166                          .DEA16:
                                        .IF DF  M$$MGE

                                          .IF DF R$$11M

004166  022700  120000                  CMP     #120000,R0      ; IS ALLOCATION FROM PARTITION SPACE ?
004172  101040                          BHI     20$             ; NO .. DE-ALLOCATE NORMALLY

                                          .IFF  ; DF R$$11M

                                        CMP     R0,#100000      ; IS ALLOCATION FROM PARTITION SPACE?
                                        BHIS    20$             ; BR IF NO - DEALLOCATE NORMALLY

                                          .IFTF ; DF R$$11M

004174  016702  000152'                 MOV     ..PDV,R2        ; GET PDV ADDRESS
004200  001437                          BEQ     30$             ; IF ZERO, DISCARD THE ALLOCATION

                                          .IFT  ; DF R$$11M

004202                                  SWSTK$  30$             ;; ENTER KERNEL MODE
004206  016204  000000G                 MOV     Z.PCB(R2),R4    ;; GET PCB ADDRESS

                                          .IFF  ; DF R$$11M

                                        JSR     R5,$SAVRG       ;; SAVE R3-R5
                                        MOV     Z.GCD(R2),R4    ;; GET GCD ADDRESS

                                          .IFTF ; DF R$$11M

004212  010003                          MOV     R0,R3           ;; COPY ALLOCATION ADDRESS
004214  010105                          MOV     R1,R5           ;; AND SIZE

                                          .IFT  ; DF R$$11M

004216  016400  000000G                 MOV     P.REL(R4),R0    ;; GET PROCESS BIAS
```


NTLUNL
SYMBOL
SYMBOL
E.LCN
E.LEN
E.LIN
E.LNK
E.MOD
E.NOD
E.PDV
E.PORT
E.PRM
E.PVC
E.SIZ
E.TIME
FC.FCT
FE.CEX
FF.ADD
FF.CIR
FF.CON
FF.FIL
FF.HST
FF.LIN
FF.MOD
FF.MON
FF.MSK
FF.PRT
FF.QUL
FF.REM
FILHD
FLT
FLT16
FL.PX1
FL.PX2
FMASK
FMKTMP
FNDADD
FR$BCC
FR$CCF
FR$CDF
FR$DAO
FR$EXC
FR$FRM
FR$FTL
FR$OPN
FR$RFD
FR$SBU
FR$SHO
FR$UBU
FR$UPT
FS.TRM
F.ADD
F.CEV
F.CIR
F.CLS

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES		
E.LCN		000042		#5-66		
E.LEN		000216		#5-66		
E.LIN		000024		#5-66		
E.LNK		000000		#5-66		
E.MOD		000036		#5-66		
E.NOD		000034		#5-66		
E.PDV		000021		#5-66		
E.PORT		000040		#5-66		
E.PRM		000026		#5-66		
E.PVC		000044		#5-66		
E.SIZ		000022		#5-66		
E.TIME		000004		#5-66		
FC.FCT	=	000030		12-488	19-1115	
FE.CEX	=	******	GX	19-1070	19-1241	
FF.ADD	=	000020		#5-67	5-67	
FF.CIR	=	000040		#5-67	5-67	
FF.CON	=	000001		#5-67		
FF.FIL	=	000002		#5-67		
FF.HST	=	040000		#5-67		
FF.LIN	=	000010		#5-67	5-67	
FF.MOD	=	000100		#5-67	5-67	
FF.MON	=	000004		#5-67		
FF.MSK	=	000077		#5-67		
FF.PRT	=	000200		#5-67	5-67	
FF.QUL	=	000370		#5-67		
FF.REM	=	100000		#5-67		
FILHD	=	******	GX	26-1537		
FLT		003352	R	#27-1572		
FLT16		003352	R	14-600	#27-1571	28-1664
FL.PX1	=	******	GX	8-228	11-439	11-448
FL.PX2	=	******	GX	11-453		
FMASK	=	******	GX	19-1070	19-1241	
FMKTMP		000002	R	#7-135		
FNDADD	=	******	GX	14-590		
FR$BCC	=	000007		#5-66		
FR$CCF	=	000001		#5-66		
FR$CDF	=	000002		#5-66		
FR$DAO	=	000011		#5-66		
FR$EXC	=	000000		#5-66		
FR$FRM	=	000010		#5-66		
FR$FTL	=	000005		#5-66		
FR$OPN	=	000004		#5-66		
FR$RFD	=	000006		#5-66		
FR$SBU	=	000012		#5-66		
FR$SHO	=	000003		#5-66		
FR$UBU	=	000013		#5-66		
FR$UPT	=	000014		#5-66		
FS.TRM	=	003000		12-488	19-1115	
F.ADD		000016		#5-67		
F.CEV		000030		#5-67		
F.CIR		000024		#5-67		
F.CLS		000002		#5-67	26-1547	

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 5

NTL11M -
Symbol t

AS.DEL=
AS.EXT=
AS.RED=
AS.WRT=
A$$CHK=
A$$CPS=
A$$PRI=
A$$TRP=
A.IOC
A.LGTH=
A.MPCT
A.PCB
A.PCBL
A.PRI
A.STAT
A.TCB
A.TCBL
CMODE =
C$$CKP=
C$$ORE=
C$$RSH=
DV.CCL=
DV.COM=
DV.DIR=
DV.EXT=
DV.F11=
DV.ISP=
DV.MBC=
DV.MNT=
DV.MSD=
DV.OSP=
DV.PSE=
DV.REC=
DV.SDI=
DV.SQD=
DV.SWL=
DV.TTY=
DV.UMD=
D$$BUG=
D$$ISK=
D$$L11=
D$$SNC=
D$$YNM=
D.DSP
D.LNK
D.MSK
D.NAM
D.PCB
D.UCB
D.UCBL
D.UNIT
D.VCAN=
D.VDEB=
D.VINI=
D.VOUT=
D.VPWF=
E$$XPR=

NTL11M -

Symbol table

```
AS.DEL= 000010          FE.CAL= 000040          I$$RAR= 000000          MPCSR  = 177746          P.PRI   000002
AS.EXT= 000004          FE.CEX= 020000 G        I$$RDN= 000000          M$$CRB= 000124          P.PRO = ****** G
AS.RED= 000001          FE.DRV= 000010          KDSAR0= 172360          M$$CRX= 000000          P.REL   000014 G
AS.WRT= 000002          FE.DYM= 010000          KDSDR0= 172320          M$$FCS= 000000          P.SIZE  000016 G
A$$CHK= 000000          FE.EXP= 000200          KINAR0= 172340          M$$MGE= 000000          P.STAT  000030 G
A$$CPS= 000000          FE.EXT= 000001 G        KINAR5= 172352          M$$NET= 000000          P.SUB   000010 G
A$$PRI= 000000          FE.EXV= 000004 G        KINAR6= 172354          M$$OVR= 000000          P.SWSZ  000022 G
A$$TRP= 000000          FE.FDT= 002000          KINAR7= 172356          M.BFVH  000011 G        P.TCB   000026
A.IOC   000003          FE.LSI= 000400          KISAR0= 172340          M.BFVL  000012 G        P.WAIT  000020 G
A.LGTH= 000014          FE.MUP= 000002          KISAR5= 172352          M.LGTH= 000014 G        Q$$OPT= 000010
A.MPCT  000011          FE.MXT= 040000          KISAR6= 172354          M.LNK   000000          R$LDAT  000026
A.PCB   000012          FE.NLG= 100000          KISAR7= 172356          M.UMRA  000002 G        R$LFLG  000024
A.PCBL  000000          FE.OFF= 001000          KISDR0= 172300          M.UMRN  000004 G        R$LHGV  000006
A.PRI   000002          FE.PKT= 000100          KISDR6= 172314          M.UMVH  000010          R$LLDZ  000012
A.STAT  000010          FE.PLA= 000020 G        KISDR7= 172316          M.UMVL  000006          R$LMXV  000010
A.TCB   000004          FE.X25= 004000          K$$CNT= 177546          N$$ACC= 000001          R$LMXZ  000014
A.TCBL  000006          F$$LVL= 000001          K$$CSR= 177546          N$$BUF= 000001          R$LNAM  000000
CMODE = 140000          F2.ACN= 000020          K$$LDC= 000000          N$$LDV= 000001          R$LOFF  000016
C$$CKP= 000000          F2.AHR= 010000          K$$TPS= 000074          N$$MCP= 000001          R$LSA   000004
C$$ORE= 000400          F2.DAS= 000001          LB$REV= 000400          N$$MLL= 000001          R$LSEG  000022
C$$RSH= 177564          F2.DPR= 000400          LD$ACC= 100000          N$$MOV= 000010          R$LSIZ  000034
DV.CCL= 000002          F2.EVT= 000010          LD$CLS= 020000          N$$NCT= 000001          R$LWND  000020
DV.COM= 020000          F2.GGF= 002000          LD$LP = 000000          N$$PEM= 000001          R$$DER= 000000
DV.DIR= 000010          F2.IRR= 001000          LD$REL= 000004          PIRQ   = 177772         R$$K11= 000001
DV.EXT= 000400          F2.LIB= 000002          LD$RSV= 040000          PMODE  = 030000         R$$SND= 000000
DV.F11= 040000          F2.MP  = 000004         LD$SUP= 000010          PR0    = 000000         R$$11M= 000000
DV.ISP= 002000          F2.POL= 000100          L$BASG  001000          PR1    = 000040         SISDR0= 172200
DV.MBC= 000400          F2.RAS= 004000          L$BBLK  000360          PR4    = 000200         SPARE = 000010
DV.MNT= 100000 G        F2.RBN= 020000          L$BDAT  000032          PR5    = 000240         SP.EIP= 000001
DV.MSD= 000100          F2.SDW= 000040          L$BEXT  000352 G        PR6    = 000300         SP.ENB= 000002
DV.OSP= 004000          F2.STP= 100000          L$BFLG  000030 G        PR7    = 000340         SP.LOG= 000004
DV.PSE= 010000          F2.SWP= 040000          L$BFL2  000774          PS     = 177776         SR0    = 177572
DV.REC= 000001          F2.WND= 000200          L$BHGV  000012          PS.APR= 000007          SR3    = 172516
DV.SDI= 000020          F3.AST= 000200          L$BHRB  000356 G        PS.CHK= 010000          SWR    = 177570
DV.SQD= 000040          F3.CLI= 001000          L$BLDZ  000016 G        PS.CKP= 040000          S$$WRG= 000000
DV.SWL= 001000          F3.CRA= 000001          L$BLIB  000040          PS.CKR= 020000          S$$YSZ= 007600
DV.TTY= 000004          F3.EIS= 000004          L$BLRL  000776          PS.COM= 000200 G        S.BMSK  177776
DV.UMD= 000200          F3.NWK= 000002          L$BLUN  000362          PS.DEL= 000010          S.BMSV  177774
D$$BUG= 177514          F3.PMN= 004000          L$BMXV  000014          PS.DRV= 000020          S.CON   000010 G
D$$ISK= 000000          F3.PRO= 000040          L$BMXZ  000020 G        PS.FXD= 004000          S.CSR   000012
D$$L11= 000001          F3.RLK= 020000          L$BOFF  000022          PS.LIO= 001000          S.CTM   000006
D$$SNC= 000000          F3.SHF= 040000          L$BPAR  000004          PS.NSF= 000400 G        S.DMCS  000020
D$$YNM= 000000          F3.STM= 000010          L$BPRI  000346          PS.OUT= 100000          S.FRK   000016
D.DSP   000012 G        F3.TCM= 002000          L$BROB  000364          PS.PER= 002000          S.ITM   000007
D.LNK   000000          F3.UDS= 000020          L$BROL  000366          PS.PIC= 000100          S.LHD   000000 G
D.MSK   000014          F3.WAT= 010000          L$BSA   000010 G        PS.SYS= 000040 G        S.PKI   000014
D.NAM   000004 G        F3.XHR= 000100          L$BSEG  000026          P$$P45= 000000          S.PRI   000004
D.PCB   000034 G        F3.11S= 000400          L$BSGL  000354          P$$WRD= 000000          S.RCNT  177772
D.UCB   000002 G        G$$TPP= 000000          L$BSYS  000025          P.ATT  = ******* G      S.ROFF  177773
D.UCBL  000010 G        G$$TSS= 000000          L$BTSK  000000          P.BLKS  000016 G        S.STS   000011
D.UNIT  000006 G        G$$TTK= 000000          L$BWND  000024          P.BUSY  000024 G        S.VCT   000005
D.VCAN= 000002          G$$WRD= 000000          L$BXFR  000350          P.IOC   000003          S1.DEC= 001000
D.VDEB= 177776          HF.CIS= 000200          L$$ASG= 000000          P.LGTH= ****** G        S1.DPR= 000400
D.VINI= 000000          HF.EIS= 000002          L$$DRV= 000000          P.LNK   000000          S1.DSI= 004000
D.VOUT= 000004          HF.FPP= 100000          L$$P11= 000001          P.MAIN  000012 G        S1.ESC= 000002
D.VPWF= 000006 G        HF.QB  = 000004         L$$11R= 000000          P.NAM   000004 G        S1.IBF= 002000
E$$XPR= 000000          HF.UBM= 000001          MPAR   = 172100         P.OWN   000026          S1.IBY= 000200
```


178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234

```
                                        .SBTTL  $XDEAC - DEALLOCATE CORE BUFFER IN EXTENDED POOL
                                ;+
                                ; **-$XDEAC-DEALLOCATE CORE BUFFER IN EXTENDED POOL
                                ;
                                ; THIS ROUTINE IS CALLED TO DEALLOCATE A CORE BUFFER IN EXTENDED POOL.
                                ; THE BLOCK IS INSERTED INTO THE FREE BLOCK CHAIN BY CORE ADDRESS. IF
                                ; AN ADJACENT BLOCK IS CURRENTLY FREE, THEN THE TWO BLOCKS ARE MERGED
                                ; AND INSERTED IN THE FREE BLOCK CHAIN.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0=ADDRESS OF THE CORE BUFFER TO BE DEALLOCATED.
                                ;       R1=SIZE OF THE CORE BUFFER TO DEALLOCATE IN BYTES.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       THE CORE BLOCK IS MERGED INTO THE FREE CORE CHAIN BY CORE
                                ;       ADDRESS AND IS MERGED IF NECESSARY WITH ADJACENT BLOCKS.
                                ;
                                ; REGISTERS:
                                ;       R2 MODIFIED
                                ;
                                ;-
000134                          $XDEAC::
000134                                  SAVRG   <R3>            ; SAVE R3
                                        .IF NDF M$$MGE

                                        CALL    $DEACB          ; DEALLOCATE TO RSX POOL IN UMAPPED SYSTEMS
                                        BR      110$            ; AND EXIT

                                        .IFF

000136  032700  000001                  BIT     #1,R0           ; IS THE BLOCK IN EXTENDED POOL ?
000142  001003                          BNE     5$              ; IF NE, YES

                                          .IF DF R$$11M
000144                                  CALL    @DEACB          ; ELSE, DEALLCCATE RSX CORE BLOCK
000150  000523                          BR      110$            ; AND EXIT

                                          .IFF  ; R$$11M

                                        MOV     R4,-(SP)        ; SAVE R4
                                        MOV     R1,R3           ; COPY SIZE OF BLOCK
                                        MOV     R0,R1           ;  AND IT'S ADDRESS

                                        .INH0                   ; INHIBIT TASK SWITCHING
                                        CALL    $$RLCB          ;; RELEASE THE BLOCK
                                        .ENB0                   ;; ENABLE TASK SWITCHING

                                        MOV     (SP)+,R4        ; RESTORE R4
                                        BR      110$            ; AND EXIT

                                          .ENDC ; DF R$$11M

000152                          5$:     SAVMAP                  ; SAVE CURRENT APR 5 MAPPING
000156  010446                          MOV     R4,-(SP)        ; SAVE SOME REGISTERS
```

```
                                        .SBTTL  $ALP16 - ALLOCATE FROM PROCESS PARTITION
                                ;+
                                ; *** - $ALP16 - ALLOCATE SINGLEWORD FROM PROCESS PARTITION
                                ;
                                ; INPUTS:
                                ;       R1 = SIZE OF BUFFER REQUIRED IN BYTES.
                                ;       $PDVA = PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT   SUCCESS/FAILURE
                                ;       R0 = BUFFER ADDRESS
                                ;       ALL OTHER REGISTERS PRESERVED
                                ;
                                ;-
000000                          $ALP16::
                                        .IF DF  R$$ M

000000                                  SWSTK$  20$             ;* ENTER KERNEL MODE
000004  016703  000000G                 MOV     $PDVA,R3        ;* GET PDV ADDRESS FOR THIS PROCESS
000010  016367  000000G 000000G         MOV     Z.AVL(R3),$PRAVL ;* GET LISTHEAD OF AVAILABLE SPACE

                                         .IF DF M$$MGE

000016  010104                          MOV     R1,R4           ;* SAVE BYTE COUNT
000020  016700  000000G                 MOV     $BIAS,R0        ;* GET RELOCATION BIAS FOR THIS PROCESS
000024                                  CALL    $MAPX           ;* MAP TO PROCESS
000030  010401                          MOV     R4,R1           ;* RESTORE BYTE COUNT
000032  012700  177776G                 MOV     #$PRAVL-2,R0    ;* GET ADDRESS OF LISTHEAD-2
000036                                  CALL    $ALOC2          ;* TRY TO GET BUFFER FROM END OF PARTITION
000042  103006                          BCC     10$             ;* IF CC, SUCCESS

                                         .IFF   ; DF M$$MGE

                                        MOV     #$PRAVL-2,R0    ;* GET ADDRESS OF LISTHEAD-2
                                        CALL    @ALOC1          ;* TRY TO GET BUFFER FROM END OF PARTITION
                                        BCC     10$             ;* IF CC, SUCCESS

                                         .IFTF  ; DF M$$MGE

000044  017700  000000G                 MOV     @HEADR,R0       ;* GET USER MODE SP
000050  011000                          MOV     (R0),R0         ;* ...
000052  005260  000006                  INC     6(R0)           ;* SET USER MODE C-BIT
000056  000412                          BR      20$             ;* LEAVE R0 UNCHANGED
000060                          10$:
                                         .IFT   ; DF M$$MGE

000060  052700  120000                  BIS     #120000,R0      ;* MAKE SURE THAT WE RE-MAP WHEN USING THIS
                                                                ;*  ADDRESS
000064  016763  000000G 000000G         MOV     $PRAVL,Z.AVL(R3) ;* RESTORE POINTER TO FREE SPACE
000072  042763  160000  000000G         BIC     #160000,Z.AVL(R3) ;* CLEAR LAST APR USED
000100  010066  000004                  MOV     R0,4(SP)        ;* RETURN BUFFER ADDRESS IN R0
                                                                ;* REMEMBER: $MAPX IS A CO-ROUTINE

                                         .IFF   ; DF M$$MGE

                                        MOV     $PRAVL,Z.AVL(R3) ;* RESTORE LISTHEAD
```

```
                                        .SBTTL  $PRIO - PROCESS I/O

                                ;+
                                ; $PRIO - PROCESS I/O
                                ;
                                ; THIS ROUTINE IS CALLED TO TRANSFER THE PROCESS TASK IMAGE INTO CORE. A PCB
                                ; WILL BE CREATED IF THE PARTITION IN USE IS A SYSTEM CONTROLLED PARTITION.
                                ; THE PRESENCE OF AN MCR PROCESS IN A SYSTEM CONTROLLED PARTITION IS DENOTED
                                ; BY SETTING THE PARTITION NAME TO "NT.***", WHERE "***" IS THE PDV NAME.
                                ;
                                ; INPUTS:
                                ;       R1=FILE SPEC ADDRESS
                                ;       R2=FILE SPEC LENGTH
                                ;       $$PAR=PARTITION NAME (OPTIONAL)
                                ;       $$INC=EXTENSION VALUE IN 32. WORD BLOCKS (OPTIONAL)
                                ;       $NAME=RAD50 PDV NAME
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=ERROR CODE
                                ;       R1=PARTITION CONTROL BLOCK ADDRESS/GCD BLOCK ADDRESS
                                ;       R2=DESTROYED
                                ;-
000000                          $PRIO::
000000  005067  000012'                CLR     PCBA            ; NO PCB YET
000004  005067  000010'                CLR     ERROR           ; NO ERROR, EITHER
000010                                 CALL    $CLOPE          ; OPEN PROCESS FILE
000014  105777  000000G               TSTB    @$CLIOS         ; LOOK AT GCL IOSB
000020  100445                        BMI     10$             ; IF MI, FAILURE
000022  052010                        BIS     (R0)+,(R0)      ; IS FILE CONTIGUOUS?
000024  001447                        BEQ     11$             ; IF EQ, NO
000026                                 CALL    PR.LBL          ; READ LABEL BLOCK
000032  103405                        BCS     10$             ; IF CS, ERROR

000034                                 CALL    PR.PAR          ; FIND PROCESS PARTITION
000040  103402                        BCS     10$             ; IF CS, ERROR
000042                          5$:    CALL    PR.XFR          ; READ PROCESS TASK IMAGE INTO CORE
000046                          10$:   CALL    $CLDEQ          ; CLOSE PROCESS FILE
000052  005767  000010'               TST     ERROR           ; ANY ERRORS? (CLEARS C-BIT)
000056  001411                        BEQ     30$             ; IF ZERO, NO

000060  016700  000012'               MOV     PCBA,R0         ; GET PCB ADDRESS
000064  001405                        BEQ     20$             ; IF ZERO, NOTHING TO CLEANUP
000066                                 CALL    $XPAR           ; RELEASE PARTITION
000072  103430                        BCS     121$            ; IF CS, VERY BAD
000074  005067  000012'               CLR     PCBA            ; SAFETY PLAY
000100  000261                 20$:   SEC                     ; INDICATE FAILURE
000102  016700  000010'         30$:   MOV     ERROR,R0        ; GET ERROR CODE

000106  016701  000012'               MOV     PCBA,R1         ; AND PCB ADDRESS

000112  103407                        BCS     40$             ; DON'T CHECK IF ERROR HAS OCCURRED
000114  032777  000000G 000000G       BIT     #FE.PLA,@FMASK  ; IS THIS A PLAS SYSTEM ?
000122  001403                        BEQ     40$             ;   NO
000124  012761  177777  000000G       MOV     #-1,P.PRO(R1)   ; NO PLAS ACCESS OF ANY SORT ALLOWED

```

```
                                        .SBTTL  PR.XFR - READ PROCESS INTO CORE
                                ;+
                                ;
                                ; PR.XFR - READ PROCESS TASK IMAGE INTO CORE
                                ;
                                ; THIS ROUTINE IS CALLED TO READ THE PROCESS TASK IMAGE INTO CORE.
                                ;
                                ; INPUTS:
                                ;       PCBA=PCB ADDRESS
                                ;       GCDA=GCB ADDRESS (IAS ONLY)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=DESTROYED
                                ;-
001620                          PR.XFR:
                                        .IF DF  R$$11M
001620  016700  000012'                 MOV     PCBA,R0         ; GET PCB ADDRESS
001624  016000  000014                  MOV     P.REL(R0),R0    ; GET PARTITION STARTING ADDRESS
                                        .IFF
                                        MOV     R5,-(SP)        ; SAVE SOME REGISTERS
                                        MOV     R4,-(SP)        ; ...
                                        MOV     GCDA,R0         ; GET GCD ADDRESS
                                        MOV     G.BA(R0),R0     ; GET BASE ADDRESS OF REGION
                                        MOV     R0,R5           ; SAVE BASE ADDRESS OF REGION
                                        .ENDC
001630                                  CALL    $READX          ; READ INTO EXTENDED MEMORY
001634  103401                         BCS     101$            ; IF CS, FAILURE
                                        .IF DF  R$$11D ! I$$AS
                                        BIT     #FL.DDM,$FLAGS  ; ARE WE LOADING A DDM?
                                        BEQ     10$             ; BR IF NO
                                        MOV     $NTLPT,R0       ; GET BIAS OF NTL HOME BLOCK
                                        CALL    $MAPX           ;; MAP TO THE HOME BLOCK
                                        MOV     .CXGCD(R2),R4   ;; GET GCD ADDRESS OF COMM EXEC
                                        CALL    @(SP)+          ;; RESTORE MAPPING
                                        MOV     R5,R0           ; GET BASE ADDRESS OF REGION
                                        CALL    $MAPX           ;; MAP TO PROCESS
                                        MOV     G.BA(R4),(R2)   ;; STORE COMM EXEC BIAS
                                        CALL    @(SP)+          ;; CALL CO-ROUTINE TO RESTORE MAPPING
                                10$:    MOV     (SP)+,R4        ; RESTORE R4
                                        MOV     (SP)+,R5        ; RESTORE R5
                                        .ENDC
001636                                  RETURN
                                ;
                                ; ERROR CONDITION
                                ;
001640  116777  0000006 0000006 101$:   MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
001646  012767  000042  000010'         MOV     #ERR34,ERROR    ; PROCESS IMAGE READ FAILURE
001654                                  RETURN
                                        .END
000001
```

```
                                        .IF DF  R$$11M

000202  052722  140000                BIS     #140000,(R2)+   ;       ....

                                        .IFF    ; DF R$$11M

                                        BIS     #40000,(R2)+    ;USE APR2 MAPPING

                                        .ENDC   ; DF R$$11M

000206  061216                          ADD     (R2),(SP)       ;CALCULATE ADDRESS OF NEXT BLOCK
000210  020026                          CMP     R0,(SP)+        ;EQUAL TO BLOCK BEING RELEASED?
000212  001003                          BNE     50$             ;IF NE NO
000214  061201                          ADD     (R2),R1         ;MERGE TWO BLOCKS
000216  011042                          MOV     (R0),-(R2)      ;MOVE LINK WORD TO PREVIOUS BLOCK
000220  010200                          MOV     R2,R0           ;SET NEW ADDRESS OF BLOCK
000222  010160  000002          50$:    MOV     R1,2(R0)        ;SET SIZE OF BLOCK RELEASED
000226  012602                          MOV     (SP)+,R2        ;RESTORE ADDRESS OF LISTHEAD
000230                          60$:    RETURN

000232  005703                  70$:    TST     R3              ;;INSERT AT END OF LIST?
000234  001756                          BEQ     40$             ;;IF EQ YES, OK
000236                          80$:    CRASH                   ;;ILLEGAL DEALLOCATION

        000001                          .END
```

```
                        .SBTTL  REP5B - REPLACE REGION NAME
                ;+
                ; REP5B - REPLACE REGION NAME
                ;
                ; INPUTS:
                ;       $RNAME=REGION NAME
                ;       R3=FORMAT STRING ADDRESS
                ;       R4=ERROR MESSAGE BUFFER ADDRESS
                ;
                ; OUTPUTS:
                ;       R0,R1,R2=DESTROYED
                ;       R3,R4=UPDATED
                ;-
                        .IF DF  R$$11D!I$$AS

                REP5B:  CALL    $REP.P          ; REPLACE PROCESS NAME
                10$:    MOVB    (R3)+,(R4)      ; LOOK FOR SECOND ASTERISK
                        CMPB    #'*,(R4)+       ;  ...
                        BNE     10$             ;  ...
                        DEC     R4              ; SKIP THE ASTERISK
                        MOV     R4,R0           ; COPY BUFFER ADDRESS
                        MOV     $RNAME,R1       ; GET FIRST HALF OF NAME
                        CALL    $C5TA           ; CONVERT TO ASCII
                        MOV     $RNAME+2,R1     ; GET SECOND HALF OF NAME
                        BEQ     20$             ; IF EQ, DON'T BOTHER
                        CALL    $C5TA           ; CONVERT TO ASCII
                20$:    CMPB    #' ,-(R0)       ; SKIP TRAILING SPACES
                        BEQ     20$             ;  ...
                        INC     R0              ;  ...
                        MOV     R0,R4           ; SET NEW BUFFER ADDRESS
                        RETURN

                        .ENDC
000001                  .END
```

```
                                        .SBTTL  $ROL    - ALLOCATE REACHIBILITY MATRIX
                                ;+
                                ; $ROL  - ALLOCATE REACHIBILITY MATRIX
                                ;
                                ; INPUTS:
                                ;       $PDVA - XPT PDV ADDRESS
                                ;       R2    - XPT DDB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       REACHIBILITY MATRIX ALLOCATED
                                ;-
                                $ROL::  CALL    $SAVAL                  ;SAVE REGS
000466 016705  000000G          MOV     $PDVA,R5                ;STORE PDV ADDRESS
000472 005004                   CLR     R4                      ;INIT REG
000476 156504  000000G          BISB    Z.LLN(R5),R4            ;STORE NUMBER OF XPT CHANNELS
000500 017703  000000G          MOV     @DECPT,R3               ;COPY DECNET HOME BLOCK ADDRESS
000504 016300  000060           MOV     D$NN(R3),R0             ;STORE MAX NODES
000510 001555                   BEQ     80$                     ;IF ENDNODE - BRANCH
000514
000516 016301  000062           MOV     D$NA(R3),R1             ;MULTIPLE AREAS ?
000522 001401                   BEQ     10$                     ;IF NO - BRANCH
000524 005200                   INC     R0                      ;INCLUDE EXTRA AREA ENTRY
000526 005200           10$:    INC     R0                      ;INCLUDE EXTRA NODE ENTRY
000530 066300  000062           ADD     D$NA(R3),R0             ;INCLUDE AREAS
000534 006300                   ASL     R0                      ;CONVERT TO BYTES
000536 010001                   MOV     R0,R1                   ;SET UP R1 FOR ALLOCATION
000540 062701  000077           ADD     #77,R1                  ;ROUND UP TO BLOCK BOUNDARY
000544 042701  000077           BIC     #77,R1                  ; ...
000550                          ASR$    6,R1                    ; CONVERT TO BLOCK COUNT
000564
000570 103004                   CALL    $ALN18                  ;ALLOCATE ROL TABLE
000572                          BCC     20$                     ;IF SUCCESS - BRANCH
                                EMSG$   9D                      ;TYPE ERROR
000600
                                RETURN
000602 010062  000024   000022   20$:    MOV     R0,N$ROA1+2(R2)         ;SET UP APR BIAS FOR NODE ROL
000606 016362  000060           MOV     D$NN(R3),N$ROA1(R2)     ;SET COUNT
000614 0C5262  000022           INC     N$ROA1(R2)              ;INCREASE FOR XPT ROL OFFSET
000620 005000                   CLR     R0                      ;CREATE APR 6 VIRTUAL ADDRESS
000622                          BIAS    R0                      ; ...
000632 010062  000026           MOV     R0,N$ROA1+4(R2)         ;SET UP VIRTUAL ADDRESS FOR NODE ROL
000636 016300  000062           MOV     D$NA(R3),R0             ;GET NUMBER OF AREAS
000642 001417                   BEQ     30$                     ;IF NONE - BRANCH
000644 016262  000024   000032   MOV     N$ROA1+2(R2),N$ROA2+2(R2) ;SET UP APR BIAS FOR AREA ROL
000652 010062  000030           MOV     R0,N$ROA2(R2)           ;SET AREA COUNT
000656 005262  000030           INC     N$ROA2(R2)              ;INCREASE FOR XPT ROL OFFSET
000662 016262  000026   000034   MOV     N$ROA1+4(R2),N$ROA2+4(R2) ;COPY START VIR. ADDRESS
000670 016200  000022           MOV     N$ROA1(R2),R0           ;GET NODE ROL COUNT
000674 006300                   ASL     R0                      ;CONVERT TO BYTES
000676 060062  000034           ADD     R0,N$ROA2+4(R2)         ;STORE AREA VIRTUAL ROL ADDRESS
000702                  30$:    SWSTK$  80$
```

```
                                        ;+
                                        ; READ - READ NEXT RECORD FROM SYMBOL TABLE FILE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT = SUCCESS/FAILURE
                                        ;       R0,R1,R2 = DESTROYED
                                        ;       R3 = RECORD LENGTH
                                        ;       R4 = RECORD ADDRESS
                                        ;-
                                                .ENABL  LSB

000222  012767  000000'  000000G READ:  MOV     #STBBF,$CLBUF   ; SET GCL BUFFER ADDRESS
000230  012767  000172   000000G        MOV     #122.,$CLSIZ    ; AND BUFFER SIZE
000236                                  CALL    $GCLEP          ; READ NEXT RECORD FROM STB FILE
000242  016700  000000G                 MOV     $CLIOS,R0       ; GET GCL IOSB ADDRESS
000246  105710                         TSTB    (R0)            ; SUCCESS ? (CLEARS C-BIT)
000250  100405                         BMI     101$            ; NO ..
000252  016003  000002                 MOV     2(R0),R3        ; GET RECORD LENGTH
000256  012704  000000'                MOV     #STBBF,R4       ; AND RECORD ADDRESS
000262                                  RETURN

                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000264  012766  000015  000010  101$:   MOV     #<<E.ORUN*2>+1>,8.(SP) ; ASSUME BUFFER TOO SMALL
000272  122710  000000G                CMPB    #IE.RBG,(R0)    ; CORRECT ASSUMPTION ?
000276  001411                         BEQ     112$            ; YES ..
000300  012766  000011  000010         MOV     #<<E.READ*2>+1>,8.(SP) ; ASSUME GENERAL READ FAILURE
000306  122710  000000G                CMPB    #IE.EOF,(R0)    ; IS IT END OF FILE ?
000312  001003                         BNE     112$            ; NO .. CORRECT ASSUMPTION
000314  012766  00002'   000010  111$:   MOV     #<<E.BAD*2>+1>,8.(SP) ; IT'S BAD FILE FORMAT
000322  000261                 112$:   SEC                     ; INDICATE FAILURE
000324                                  RETURN
```

```
                                ;+
                                ; $DVCHA - ".DVCHA" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000230                          $DVCHA::
000230                                  CALL    $VXLLC          ; MAKE SURE IT'S NOT AN LLC
000234                                  CALL    $RQ1            ; REQUEST BINARY FILE
000240  103405                          BCS     10$             ; IF CS, NO ACCESS
000242  112712  000000G                 MOVB    #.DVCHA,(R2)    ; STORE FUNCTION CODE
000246  062767  000004  000000G         ADD     #4,$SIZE        ; ADJUST DATA BASE SIZE
000254                          10$:    RETURN
```

```
                                        ;+
                                        ; $NUM - NUMERIC EXPRESSION ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       .PNUMH=HIGH ORDER NUMERIC VALUE
                                        ;       .PNUMB=LOW ORDER NUMERIC VALUE
                                        ;       F2.NEG SET IF NEGATIVE NUMBER
                                        ;       F2.BYT SET IF BYTE EXPRESSION
                                        ;
                                        ; OUTPUTS:
                                        ;       $VALUE=NUMERIC VALUE
                                        ;       F2.NEG CLEARED
                                        ;-
001154                                  $NUM::
001154  016767  000000G 000000G         10$:    MOV     .PNUMB,$VALUE   ; GET NUMERIC VALUE
001162  005767  000000G                 20$:    TST     .PNUMH          ; SINGLE PRECISION NUMBER?
001166  001025                          30$:    BNE     30$             ; IF NE, NO - ERROR
001170  032767  0000 0G 000000G                 BIT     #F2.BYT,$FLAG2  ; BYTE EXPRESSION?
001176  001412                                  BEQ     10$             ; IF EQ, NO
001200  105767  000001G                         TSTB    $VALUE+1        ; IS VALUE CONTAINED IN 8 BITS?
001204  001016                                  BNE     30$             ; IF NE, NO - ERROR
001206  032767  000000G 000000G                 BIT     #F2.NEG,$FLAG2  ; NEGATIVE NUMBER?
001214  001415                                  BEQ     40$             ; IF EQ, NO
001216  105467  000000G                         NEGB    $VALUE          ; NEGATE IT
001222  000406                                  BR      20$
001224  032767  000000G 000000G 10$:    BIT     #F2.NEG,$FLAG2  ; NEGATIVE NUMBER?
001232  001406                                  BEQ     40$             ; IF EQ, NO
001234  005467  000000G                         NEG     $VALUE          ; NEGATE IT
001240  100403                          20$:    BMI     40$             ; IF MI, NO LOSS OF SIGNIFICANCE
001242                                  30$:    EMSG$   18              ; PRINT TRUNCATION WARNING
001250  042767  000000G 000000G 40$:    BIC     #F2.NEG,$FLAG2  ; CLEAR NEGATIVE FLAG
001256                                          RETURN
```

```
                                        .SBTTL  $XPRO - CLEAN UP PROCESS

                                ;+
                                ; $XPRO - CLEANUP AFTER AN UNSUCCESSFUL "SET PROCESS" (RSX11M ONLY)
                                ;
                                ; THIS ROUTINE IS CALLED IF A "SET PROCESS" FAILS EITHER IN SETTING UP THE
                                ; LLC LINE TABLE OR IN SETTING UP THE RSX11M DCB DATA STRUCTURES. THE PROCESS
                                ; WILL BE RELEASED FROM ITS PARTITION AND, IF IT IS AN LLC, ITS LINE TABLE WILL
                                ; BE DE-ALLOCATED.
                                ;
                                ; INPUTS:
                                ;       $PDVA = PDV ADDRESS
                                ;       $PCB = PCB ADDRESS
                                ;       FL.PX1 OF $FLAGS SET IF PROCESS WAS TRANSFERRED INTO CORE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;-
                                        .ENABL  LSB

000000                          $XPRO::
000000  016704  000000G                MOV     $PDVA,R4        ; GET PDV ADDRESS
000004  032767  000000G 000000G        BIT     #FL.PX1,$FLAGS  ; WAS PROCESS TRANSFERRED?
000012  001422                         BEQ     55$             ; IF EQ, NO
000014  016700  000000G                MOV     $PCB,R0         ; GET PCB ADDRESS
000020  000411                         BR      5$
                                ;
000022  032764  000000G 000000G XPRO:  BIT     #ZF.INI,Z.FLG(R4) ; INIT DISPATCH NEEDED ?
000030  001403                         BEQ     1$              ; IF NO - BRANCH
000032  016700  000000G                MOV     $PCB,R0         ; GET SAVED PCB ADDRESS
000036  000402                         BR      5$              ; CONTINUE

000040  016400  000000G          1$:   MOV     Z.PCB(R4),R0    ; GET PCB ADDRESS
000044  001532                   5$:   BEQ     101$            ; IF NO PCB, LOGIC ERROR
000046  005064  000000G                CLR     Z.PCB(R4)       ; INDICATE PROCESS UNLOADED
000052                                 CALL    $XPAR           ; RELEASE PROCESS FROM PARTITION
000056  103526                         BCS     111$            ; IF CS, VERY BAD

000060                           55$:  .IF DF  R$$MPL

                                        BITB    #ZF.DDM,Z.FLG(R4) ; IS THIS A DDM ?
                                        BEQ     10$             ; IF EQ, NO
                                        CALL    FNCTB           ; FIND CTB ADDRESS FROM CETAB.STB

                                        SWSTK$  8$              ;; ENTER KERNEL MODE
                                        MOV     CTLST,R0        ;; GET START OF CTB LIST
                                 6$:    MOV     R0,R2           ;; SAVE CTB ADDRESS
                                        MOV     (R0),R0         ;; GET NEXT CTB ADDRESS
                                        BEQ     7$              ;; IF EQ, END OF LIST
                                        CMP     R0,R1           ;; POINTING AT OUR CTB ?
                                        BNE     6$              ;; IF NE, NO .. KEEP LOOKING
                                        MOV     (R0),(R2)       ;; UNLINK CTB FROM CTB CHAIN
                                 7$:    RETURN                  ;; RETURN TO USER MODE

                                 8$:    CALL    $HRCIN          ; RE-INITIALIZE HRC...

                                        .ENDC
```

```
                                        BIS     #120000,R0      ;; BIAS VECTOR ADDRESS TO USE APR5
                                        BIS     #120000,R1      ;;  ...
                                5$:     SUB     #2,R5           ;; DONE FOR ALL PROCESSORS ?
                                        BLT     7$              ;; IF LE, YES .. DIDN'T FIND UNIBUS RUN
                                        MOV     URMTB,URMTMP    ;; GET $URMTB ADDRESS
                                        ADD     R5,URMTMP       ;;  ...
                                        BIT     K.URM(R4),@URMTMP ;; CONTROLLER ON THIS CPU ?
                                        BEQ     5$              ;; IF EQ, NO .. KEEP LOOKING
                                        MOV     K6TAB,K6TMP     ;; GET $K6TAB ADDRESS
                                        ADD     R5,K6TMP        ;;  ...
                                        MOV     @K6TMP,@KSAR5   ;; MAP TO CPU PARTITION
                                6$:     CALL    ICBSC           ;; IS THIS A VALID ICB ?
                                        BCC     30$             ;; IF CC, YES .. RESET NONSENSE INTERRUPTS
                                        CLR     R0              ;; WHAT DO YOU MEAN THIS ISN'T AN ICB ???
                                        CALLR   45$             ;; LEAVE WITH ERROR
                                ;
                                ; CALCULATE LENGTH OF ICB FROM 1ST ICB IN CHAIN
                                ;
                                7$:     MOV     K.PRM(R4),R0    ;; GET ADDRESS OF 1ST ICB
                                        BEQ     44$             ;; IF EQ, NO ICB'S
                                        MOV     R4,-(SP)        ;; SAVE KRB ADDRESS
                                        CLR     -(SP)           ;; ASSUME THE ICB IS NOT FOR A CPU PARTITION
                                        CMPB    #1,@NCPU        ;; ICB POINTER CONTAIN A CPU NUMBER ?
                                        BHIS    8$              ;; IF HIS, NO
                                        MOV     #120000,(SP)    ;; GET BASE ADDRESS OF APR5
                                        MOV     R0,R4           ;; FORM CPU NUMBER IN UPPER FOUR BUTS OF R2
                                        BIC     #7777,R0        ;;  ...
                                        BIC     R0,R4           ;; CLEAN CPU NUMBER OUT OF POINTER
                                        SWAB    R0              ;; FORM CPU NUMBER;2 OF POOL FOR THE ICB
                                        .REPT   3
                                        ASR     R0              ;       ...
                                        .ENDR
                                        MOV     K6TAB,K6TMP     ;; GET $K6TAB ADDRESS
                                        ADD     R0,K6TMP        ;;  ...
                                        MOV     @K6TMP,@KSAR5   ;; MAP TO THE CPU PARTITION
                                8$:     ASL     R4              ;; FORM VIRTUAL ADDRESS OF ICB
                                        BIS     (SP)+,R4        ;;  ...
                                        CALL    ICBLEN          ;; SCAN ICB FOR LENGTH
                                        MOV     (SP)+,R4        ;; RESTORE KRB ADDRESS
                                        BCC     40$             ;; IF CC, FOUND LENGTH
                                        CLR     R0              ;; SOMETHING'S ROTTEN IN DENMARK
                                        BR      45$             ;; AND TELL SOMEONE

                                        .ENDC   ; DF    R$$MPL


                                                                ;; 11M OR DATA SPACE NOT ENABLED ON M-PLUS
001506  021002                 10$:    CMP     (R0),R2         ;; DOES VECTOR POINT AT OUR LINE TABLE?
001510  103402                         BLO     20$             ;; IF LO, NO
001512  021003                         CMP     (R0),R3         ;; DITTO?
001514  103405                         BLO     30$             ;; IF LO, YES
001516                          20$:
                                          .IF DF M$$MGE

001516                                  CALL    ICBSC           ;; SCAN FOR ICB
001522  103002                         BCC     30$             ;; IF CC, FOUND ONE

                                          .ENDC
```

```
                                        .SBTTL  XDCB - UNLOAD DEVICES

                                ;+
                                ;
                                ; XDCB - UNLOAD DEVICES WHICH ARE ALSO NETWORK PROCESSES
                                ;
                                ; INPUTS:
                                ;       R4 - PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       THE DEVICE ASSOCIATED WITH PDV IS UNLOADED
                                ;
                                ;-

003046  032764  000000G 000000G XDCB:   BIT     #ZF.INI,Z.FLG(R4) ; ZF.INI PROCESS ?
003054  001003                         BNE     5$                ; IF YES - BRANCH
003056  016467  000000G 000000G        MOV     Z.PCB(R4),$PCB    ; COPY PCB ADDRESS
003064  016700  000000G          5$:    MOV     DEVHD,R0          ; GET RSX11M DEVICE POINTER

003070                                  SWSTK$  20$               ;; ENTER KERNEL MODE
003074  011000                  10$:    MOV     (R0),R0           ;; GET NEXT DEVICE IN LIST
003076  001411                         BEQ     20$               ;; IF ZERO, ALL DONE
003100  026067  000000G 000000G        CMP     D.PCB(R0),$PCB    ;; SAME PCB AS OUR PROCESS?
003106  000241                         CLC                       ;; CLEAR C-BIT
003110  001371                         BNE     10$               ;; IF NE, NO
003112  005060  000000C                CLR     D.DSP(R0)         ;; MARK DEVICE AS "UNLOADED"
003116  005060  000000G                CLR     D.PCB(R0)         ;;   ...
003122                          20$:    RETURN                    ;; BACK TO USER MODE AND THEN TO CALLER
```

```
1840                                        .IFF    ; DF R$$11M
1841
1842                                        MOV     G.BA(R4),R0     ;; GET PROCESS BIAS
1843
1844                                        .IFTF   ; DF R$$11M
1845
1846 004222  010204                         MOV     R2,R4           ;; COPY PDV ADDRESS
1847 004224                                 CALL    $MAPX           ;; MAP TO THE PROCESS
1848 004230  010300                         MOV     R3,R0           ;; RESTORE ALLOCATION ADDRESS
1849 004232  010501                         MOV     R5,R1           ;; AND SIZE
1850 004234  012703  1777776G               MOV     #$PRAVL-2,R3    ;; POINT AT COMMON LISTHEAD
1851 004240  062704  000000G                ADD     #Z.AVL,R4       ;; POINT AT PROCESS ALLOCATION WORD
1852 004244  011463  000002                 MOV     (R4),2(R3)      ;; SETUP PARTITION FREE SPACE POINTER
1853 004250  042700  160000                 BIC     #160000,R0      ;; CLEAR GIVEN APR
1854 004254  050200                         BIS     R2,R0           ;; USE APR FROM MAPX
1855 004256                                 CALL    DEAC2           ;; DE-ALLOCATE BACK TO PARTITION SPACE
1856 004262  016714  000000G                MOV     $PRAVL,(R4)     ;; RESTORE PCB ALLOCATION WORD
1857 004266  042714  160000                 BIC     #160000,(R4)    ;;      ...
1858
1859                                        .IFF    ; DF R$$11M
1860
1861                                        CALL    @(SP)+          ;; CALL CO-ROUTINE TO RESTORE PREVIOUS MAPPING
1862
1863                                        .ENDC   ; DF R$$11M
1864
1865 004272                                 RETURN                  ;; BACK TO USER MODE
1866
1867                                        .IFF    ; DF M$$MGE
1868
1869                                        CMP     @EXSIZ,R0       ; IS ALLOCATION FROM RSX POOL ?
1870                                        BHI     20$             ; YES .. DE-ALLOCATE NORMALLY
1871                                        JSR     R5,$SAVRG       ; SAVE R3,R4,R5
1872                                        MOV     @NTLPT,R2       ; GET NTL DATA BLOCK ADDRESS
1873                                        MOV     .CXPCB(R2),R2   ; COMM EXEC PCB ADDRESS
1874                                        MOV     P.REL(R2),R3    ; AND COMM EXEC STARTING ADDRESS
1875                                        CMP     R0,R3           ; IS ALLOCATION FROM CEX POOL ?
1876                                        BLO     10$             ; NO .. IT'S FROM PARTITION SPACE
1877                                        ADD     P.SIZE(R2),R3   ; GET COMM EXEC ENDING ADDRESS
1878                                        CMP     R0,R3           ; IS ALLOCATION FROM CEX POOL ?
1879                                        BLO     20$             ; YES .. DE-ALLOCATE NORMALLY
1880                                10$:    MOV     ..PDV,R4        ; GET PDV ADDRESS
1881                                        BEQ     30$             ; IF ZERO, DISCARD THE ALLOCATION
1882                                        MOV     #$PRAVL-2,R3    ; POINT AT COMMON LISTHEAD
1883                                        ADD     #Z.AVL,R4       ; POINT AT PROCESS ALLOCATION WORD
1884                                        MOV     (R4),2(R3)      ; SETUP PARTITION FREE SPACE POINTER
1885                                        CALL    $DEAC1          ; DE-ALLOCATE BACK TO PARTITION SPACE
1886                                        MOV     $PRAVL,(R4)     ; RESTORE PCB ALLOCATION WORD
1887                                        BR      30$
1888
1889                                        .ENDC   ; DF M$$MGE
1890
1891 004274                                20$:    CALL    $DEA16          ; DE-ALLOCATE NORMALLY
1892 004300                                30$:    RETURN
```



```
NTLUNL

SYMBOL

SYMBOL

F.EVT
F.FLG
F.LEN
F.LIN
F.LNK
F.MOD
F.NXT
F.REM
F.SEV
F.USE
F.VAR
GETIND
GETPDV
ICBLEN
ICBSC
ICBSZ
INIFL
I$$AS
KMCB
KSAR5
K.VCT
K6TMP
LF.ACT
LF.BRO
LF.BWT
LF.ENA
LF.LPB
LF.MDC
LF.MFL
LF.MTP
LF.PAC
LF.ROY
LF.REA
LF.SER
LF.TIM
LF.UNL
LF.X2P
LGDDB
LGPDV
LGSTT
LLCAL
LN.CLO
LN.DUM
LN.LOA
LN.LOO
LN.OAU
LN.OFF
LN.ON
LN.OOP
LN.OPE
LN.REF
LN.SER
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 5

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL		VALUE		REFERENCES							
F.EVT		000004		#5-67							
F.FLG		000014		#5-67							
F.LEN		000016		#5-67							
F.LIN		000020		#5-67							
F.LNK		000000		#5-67							
F.MOD		000022		#5-67							
F.NXT		000000		#5-75							
F.REM		000026		#5-67							
F.SEV		000004		#5-67							
F.USE		000002		#5-75	32-2037						
F.VAR		000003		#5-75							
GETIND	=	******	GX	8-306							
GETPDV	=	******	GX	14-571	14-606	14-643					
ICBLEN		001720	R	16-926	#17-956						
ICBSC		001662	R	15-776	#16-911						
ICBSZ		000006	R	#7-137	*15-698	15-860	*17-1003				
INIFL	=	000001		#6-123	9-352						
I$$AS	=	******		10-184	15-792	27-1645	28-1777				
KMCB		000116	R	#7-163							
KSAR5	=	******	GX	24-1477	24-1495	26-1540	26-1551	31-1968	31-1984	32-2025	32-2051
K.VCT		000001		14-587	15-694						
K6TMP		000000	R	#7-134							
LF.ACT	=	100000		#5-72	14-568						
LF.BRO	=	000400		#5-72							
LF.BWT	=	000007		#5-72							
LF.ENA	=	002000		#5-72							
LF.LPB	=	001000		#5-72							
LF.MDC	=	000100		#5-72							
LF.MFL	=	004000		#5-72							
LF.MTP	=	000020		#5-72							
LF.PAC	=	000200		#5-72							
LF.RDY	=	040000		#5-72	14-568	19-1095					
LF.REA	=	010000		#5-72							
LF.SER	=	000040		#5-72							
LF.TIM	=	000010		#5-72							
LF.UNL	=	020000		#5-72	14-656						
LF.X2P	=	000000		#5-72							
LGDDB	=	******	GX	8-280	8-290						
LGPDV	=	******	GX	8-278							
LGSTT	=	******	GX	8-279							
LLCAL	=	******	GX	9-347							
LN.CLO	=	000000		#5-72							
LN.DUM	=	000005		#5-72							
LN.LOA	=	000004		#5-72							
LN.LOO	=	000003		#5-72							
LN.OAU	=	000003		#5-72							
LN.OFF	=	000001		#5-72							
LN.ON	=	000000		#5-72							
LN.OOP	=	000004		#5-72							
LN.OPE	=	000001		#5-72							
LN.REF	=	000002		#5-72							
LN.SER	=	000002		#5-72							

NTL11M -
Symbol t

S1.OBY=
S1.PTH=
S1.RES=
S1.RNE=
S1.RNF=
S1.RSP=
S1.RST=
S1.TNE=
S1.TSY=
S1.USI=
S2.BEL=
S2.BRQ=
S2.CR =
S2.CTO=
S2.CTS=
S2.ELF=
S2.FLF=
S2.IRQ=
S2.OBF=
S2.ORQ=
S2.PCU=
S2.RCU=
S2.SRQ=
S2.WAL=
S2.WRA=
S2.WRB=
S3.ACR=
S3.CTC=
S3.FDX=
S3.ICE=
S3.MHE=
S3.NEC=
S3.PPT=
S3.PTH=
S3.RAL=
S3.RES=
S3.RUB=
S3.TAB=
S3.TME=
S3.TSY=
S3.8BC=
S4.ABD=
S4.ANI=
S4.AVO=
S4.BLK=
S4.DEC=
S4.DLO=
S4.EDT=
S4.HFF=
S4.HFL=
S4.HHT=
S4.HSY=
S4.RGS=
S4.SFC=
S4.VFL=
S5.ABP=
S5.BCC=

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 6

NTL11M

Symbol table

```
S1.OBY= 000100          S5.DAO= 040000          T.PRI   000002          UD.800= 000003          U.CTL   000004 G
S1.PTH= 000010          S5.HPC= 000014          T.RCVL  000012          UISAR0= 177640          U.CTYP  000052
S1.RES= 010000          S5.HPO= 000020          T.RRFL  000072          UISAR4= 177650          U.CW1   000010 G
S1.RNE= 000020          S5.ITI= 000100          T.SAST  000054          UISAR5= 177652          U.CW2   000012
S1.RNF= 020000          S5.OXF= 000040          T.SRCT  000071          UISAR6= 177654          U.CW3   000014
S1.RSP= 000004          S5.RPO= 002000          T.STAT  000032          UISAR7= 177656          U.CW4   000016
S1.RST= 000001          S5.SW1= 000001          T.ST2   000034          UISDR0= 177600          U.CYL   = 000104
S1.TNE= 040000          S5.TMM= 000002          T.ST3   000036          UISDR4= 177610          U.DCB   000000 G
S1.TSY= 000040          S5.VER= 010000          T.TCBL  000030          UISDR5= 177612          U.FCDE= 000042
S1.USI= 100000          S5.XOF= 000004          T.TIO   000057          UISDR6= 177614          U.FNUM= 000040
S2.BEL= 020000          S5.XON= 000010          T.TKSZ  000060          UISDR7= 177616          U.GRP   = 000102
S2.BRQ= 000020          S6.EIO= 000400          T.UCB   000026          UM.CLI= 000036          U.KCSR= 000032
S2.CR  = 002000         S6.RDI= 100000          T2$CLI= 000001          UM.DSB= 000200          U.KCS6= 000034
S2.CTO= 040000          S6.RLU= 001000          T2$FMP= 000002          UM.NBR= 000400          U.LUIC  177774
S2.CTS= 100000          TPS    = 177564         T2.ABO= 000100          UM.OVR= 000001          U.MEDI= 000040
S2.ELF= 001000          TS$ACP= 020000          T2.AST= 100000          US.AJO= 000001          U.MLUN= 000050
S2.FLF= 000400          TS$CHK= 000100          T2.CAF= 000400          US.BSP= 000002          U.MUP   177772
S2.IRQ= 000200          TS$CMP= 000200          T2.CHK= 020000          US.BSY= 000200 G        U.OWN   177776
S2.OBF= 004000          TS$IOP= 000020          T2.CKD= 010000          US.FOR= 000040          U.RBNS= 000112
S2.ORQ= 000100          TS$NEW= 000001          T2.DST= 040000          US.FRK= 000002          U.RCTC= 000113
S2.PCU= 010000          TS$NHD= 040000 G        T2.FXD= 002000          US.KPF= 000001          U.RCTS= 000110
S2.RCU= 000001          TS$NSD= 002000          T2.HLT= 000200          US.LAB= 000004          U.RED   000002 G
S2.SRQ= 000040          TS$NXH= 000002          T2.REX= 001000          US.MDE= 000002          U.RED2  000034
S2.WAL= 000010          TS$PIC= 100000          T2.SEF= 004000          US.MDM= 000020          U.SCB   000020 G
S2.WRA= 000006          TS$PMD= 010000          T2.SPN= 000004          US.MNT= 000100 G        U.SHST= 000076
S2.WRB= 000002          TS$PRV= 000400          T2.STP= 000020          US.OFL= 000001          U.SHUN= 000074
S3.ACR= 000001          TS$RES= 000040          T2.WFR= 000001          US.PUB= 000004          U.SPC  = 000036
S3.CTC= 000004          TS$SLV= 004000          T3.ACP= 100000          US.PWF= 000010          U.STS   000005 G
S3.FDX= 000200          TS$SUP= 000010          T3.CAL= 000100          US.RED= 000002          U.ST2   000007
S3.ICE= 001000          TS$XHR= 000004          T3.CLI= 001000          US.SHR= 000001          U.SUB  = 000036
S3.MHE= 000400          TS.BLK= 171700          T3.GFL= 000010          US.SPU= 000002          U.TCHP  000042
S3.NEC= 000020          TS.CKP= 000200          T3.MCR= 004000          US.UMD= 000010          U.TCVP  000043
S3.PPT= 020000          TS.CKR= 000100          T3.NET= 000020          US.VV  = 000001          U.TFLK  000040
S3.PTH= 004000          TS.EXE= 100000          T3.NSD= 000200          US.WCK= 000010          U.TFRQ  000036
S3.RAL= 000010          TS.HLD= 002000          T3.PMD= 040000          UU.ABO= 000400          U.TIXL  000060
S3.RES= 010000          TS.MSG= 020000          T3.PRV= 010000          UU.ATN= 000100          U.TLPP  000053
S3.RUB= 040000          TS.NRP= 010000          T3.REM= 020000          UU.AVN= 000004          U.TMTI  000047
S3.TAB= 000002          TS.OUT= 000400          T3.ROV= 000040          UU.GUS= 000010          U.TRCK= 000100
S3.TME= 002000          TS.RDN= 040000          T3.RST= 000400          UU.IOS= 002000          U.TSTA  000026
S3.TSY= 000040          TS.RUN= 004000          T3.SLV= 002000          UU.ONL= 000020          U.TST5  000054
S3.8BC= 000100          TS.STP= 001000          T3.SWS= 000002          UU.RCT= 000002          U.TST6  000056
S4.ABD= 100000          T$$KMG= 000000          UBMPR = 170200          UU.RDY= 000200          U.TTAB  000050
S4.ANI= 000400          T$$MIN= 000000          UC.ALG= 000200          UU.SER= 000001          U.TTYP  000046
S4.AVO= 001000          T.ACTL  000052          UC.ATT= 000010          UU.SIO= 001000          U.TUX   000024
S4.BLK= 002000          T.ASTL  000016          UC.KIL= 000004          UU.SPC= 000040          U.UHVR= 000107
S4.DEC= 004000          T.ATT   000062          UC.LGH= 00000           U.ACB   000062          U.UIC   000044
S4.DLO= 000100          T.CPCB  000004          UC.NPR= 00010           U.ACP  = 000032         U.UNFL= 000052
S4.EDT= 010000          T.DPRI  000040          UC.PWF= 00002           U.ADMA  000066          U.UNIT  000006 G
S4.HFF= 000020          T.EFLG  000022          UC.QUE= 000040          U.AFLG  000064          U.UNSZ= 000114
S4.HFL= 000007          T.IOC   000003          UDSAR0= 177660          U.ATT   000022          U.UNTI= 000060
S4.HHT= 000040          T.LBN   000041          UDSDR0= 177620          U.BPKT= 000044          U.USVR= 000106
S4.HSY= 000200          T.LDV   000044 G        UD.UNS= 000000          U.BUF   000024          U.UTIL= 000036
S4.RGS= 020000          T.LNK   000000          UD.160= 000004          U.CBF  = 000032         U.VCB  = 000034
S4.SFC= 040000          T.MXSZ  000050          UD.200= 000001          U.CLI   177772          U.VSER= 000120
S4.VFL= 000010          T.NAM   000006          UD.556= 000002          U.CNT   000030          U.2MED= 000070
S5.ABP= 100000          T.OFF   000066          UD.625= 000005          U.COTQ  000030          U2.AT.= 000020
S5.BCC= 020000          T.PCB   000046          UD.8K  = 000006          U.CTCB  000026          U2.CRT= 002000
```



```
POLAL -
$XDEAC -
```

```
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
```

```
POLAL -
```

```
                                                MOV     R5,-(SP)        ; ...
                                                MOV     R0,R5           ; COPY UNMAPPED ADDRESS
                                                MOV     R0,-(SP)        ; SET UP ADDRESS FOR $CEACX
                                                CALL    $CEACX          ; MAP EXTENDED ADDRESS
                                                MOV     (SP)+,R4        ; COPY MAPPED ADDRESS
                                                SAVMAP                  ; SAVE BLOCK MAPPING
                                                MOV     #XAVLL-2,R3     ; POINT TO ALLOCATION MASK WORD
                                                ADD     (R3),R1         ; ROUND TO NEXT BOUNDARY
                                                BIC     (R3)+,R1        ; CLEAR EXCESS
                                                BEQ     100$            ; IF EQ, NOTHING TO RETURN
                                                MOV     @XAVLL,(R3)     ; COPY CEX ALLOCATION LISTHEAD
                                                TST     -(SP)           ; RESERVE A WORD ON THE STACK

                                        10$:    MOV     R3,(SP)         ; SAVE UNMAPPED ADDRESS IN RESERVED WORD
                                                MOV     R3,-(SP)        ; COPY ADDRESS OF CURRENT BLOCK
                                                CALL    $CEACX          ; MAP ADDRESS OF BLOCK
                                                MOV     (SP)+,R2        ; SAVE MAPPED ADDRESS
                                                MOV     (R2),R3         ; GET ADDRESS OF NEXT BLOCK
                                                BEQ     20$             ; IF EQ END OF CHAIN
                                                CMP     R5,R3           ; BLOCK GO BEFORE HERE?
                                                BHIS    10$             ; IF HIS NO

                                        20$:    SAVMAP                  ; SAVE CURRENT MAPPING
                                                MAP     4(SP)           ; MAP BLOCK BEING RELEASED
                                                MOV     R3,(R4)         ; ASSUME BLOCKS NOT ADJACENT
                                                RESMAP                  ; RESTORE MAPPING
                                                MOV     R5,-(SP)        ; CALCULATE ADDRESS OF NEW BLOCK
                                                ADD     R1,(SP)         ;
                                                CMP     R3,(SP)+        ; EQUAL TO NEXT IN CHAIN?
                                                BNE     30$             ; IF NE NO
                                                SAVMAP                  ; ELSE, SAVE MAPPING
                                                MOV     R3,-(SP)        ; COPY BLOCK ADDRESS
                                                CALL    $CEACX          ; MAP BLOCK
                                                MOV     (SP)+,R0        ; GET MAPPED ADDRESS
                                                MOV     (R0)+,-(SP)     ; SAVE LINK WORD
                                                ADD     (R0),R1         ; MERGE TWO BLOCKS
                                                MAP     6(SP)           ; MAP BLOCK BEING RELEASED
                                                MOV     (SP)+,(R4)      ; MOVE LINK WORD TO BLOCK RELEASED
                                                RESMAP                  ; RESTORE MAPPING
                                        30$:                            ; ADDRESS OF PREVIOUS BLOCK IS ON TOP OF STACK
                                                MOV     R5,(R2)+        ; ASSUME NO AGLOMERATION
                                                ADD     (R2),(SP)       ; CALCULATE ADDRESS OF NEXT BLOCK
                                                CMP     R5,(SP)+        ; EQUAL TO BLOCK BEING RELEASED?
                                                BNE     40$             ; IF NE NO
                                                ADD     (R2),R1         ; MERGE TWO BLOCKS
                                                SAVMAP                  ; SAVE CURRENT MAPPING
                                                MAP     2(SP)           ; MAP TO BLOCK BEING RELEASED
                                                MOV     (R4),-(SP)      ; SAVE LINK WORD
                                                MAP     2(SP)           ; RESTORE MAPPING
                                                MOV     (SP)+,-(R2)     ; MOVE LINK WORD TO PREVIOUS BLOCK
                                                MOV     (SP)+,(SP)      ; REPLACE BLOCK MAPPING
                                                MOV     R2,R4           ; SET NEW ADDRESS OF BLOCK
                                        40$:    MAP     (SP)            ; MAP TO BLOCK RELEASED
                                                MOV     R1,2(R4)        ; SET SIZE OF BLOCK RELEASED
                                                MOV     XAVLL,@XAVL     ; UPDATE CEX COPY OF LISTHEAD

                                        100$:   TST     (SP)+           ; CLEAN UP STACK
```

```
                                        MOV     R0,2(SP)        ;* RETURN BUFFER ADDRESS IN R0

                                         .ENDC  ; DF M$$MGE

000104                          20$:    RETURN                  ;* RETURN TO USER STATE AND THEN TO CALLER

                                        .IFF    ; DF R$$11M

                                        MOV     R3,-(SP)        ; SAVE SOME
                                        MOV     R4,-(SP)        ;  REGISTERS
                                        MOV     R1,-(SP)        ; SAVE BYTE COUNT
                                        MOV     $PDVA,R3        ; GET PDV ADDRESS FOR THIS PROCESS
                                        SEC                     ; ASSUME NO AVAILABLE SPACE
                                        MOV     Z.AVL(R3),$PRAVL; SET UP LISTHEAD OF AVAILABLE SPACE
                                        BEQ     30$             ; BR IF NO AVAILABLE SPACE
                                        CLC                     ;
                                        CLR     -(SP)           ; SET UP COMPLETION FLAG
                                        MOV     R1,R3           ; COPY SIZE OF BLOCK REQUIRED
                                        MOV     $BIAS,R0        ; SET UP BIAS OF PROCESS
                                        CALL    $MAPX           ;; MAP TO THE PROCESS
                                        BIS     R2,$PRAVL       ;; SET PROPER APR BITS
                                        CALL    $$RQCB          ;; ALLOCATE THE BLOCK
                                        ADC     2(SP)           ;; SET C-BIT FOR RETURN
                                        MOV     $PDVA,R3        ;; GET PDV ADDRESS FOR PROCESS
                                        MOV     $PRAVL,Z.AVL(R3);; RESET ALLOCATION POINTER
                                        BIC     #160000,Z.AVL(R3) ;; CLEAR MAPPING BITS
                                        CALL    @(SP)+          ;; CO-ROUTINE RETURN TO RESTORE MAPPING
                                        MOV     R4,R0           ; COPY ADDRESS OF ALLOCATED BLOCK
                                        ROR     (SP)+           ; SET C-BIT FOR RETURN
                                30$:    MOV     (SP)+,R1        ; RESTORE BYTE COUNT
                                        MOV     (SP)+,R4        ; RESTORE
                                        MOV     (SP)+,R3        ;  REGISTERS
                                        RETURN

                                        .ENDC   ; DF R$$11M
```

```
    253 000132                          40$:    RETURN
    254
    255                                 ;
    256                                 ; ERROR CONDITIONS
    257                                 ;
    258 000134  012767  000002  000010' 101$:   MOV     #ERR2,ERROR     ; OPEN FAILURE
    259 000142  000756                          BR      20$
    260 000144  012767  000006  000010' 111$:   MOV     #ERR6,ERROR     ; NOT CONTIGUOUS
    261 000152  000735                          BR      10$
    262 000154                          121$:   CRASH                   ; SOMEBODY UNLINKED OUR SUB-PCB!
```


ALOCB
AS.DE
AS.EX
AS.RE
AS.WR
A$$CHI
A$$CP
A$$PR
A$$TR
A.IOC
A.LGT
A.MPC
A.PCB
A.PCB
A.PRI
A.STA
A.TCB
A.TCB
BLOX
BUFUM
BYTES
CMODE
C$LEN
C$NID
C$NXH
C$TIM
C$$CK
C$$OR
C$$RS
DEACB
DECPT
DSKBU
D$AMX
D$AMX
D$ANN
D$BRP
D$BRT
D$DEL
D$DEL
D$END
D$FNE
D$HIC
D$HOS
D$INA
D$INC
D$IPL
D$LID
D$LNA
D$LNU
D$LST
D$MA
D$MA
D$MA
D$MLL
D$MNC
D$NA
D$NBE

Symbol table

```
ALOCB = ****** GX       D$NBRA  000054          LB$REV= 000400          N$DLY   000014          PS.SYS= 000040
AS.DEL= 000010          D$NEND= 000054          LD$ACC= 100000          N$ELEN  000054          P$$P45= 000000
AS.EXT= 000004          D$NLN   000030          LD$CLS= 020000          N$ENC   000042          P$$WRD= 000000
AS.RED= 000001          D$NN    000060          LD$LP = 000000          N$ERRC  000022          P.BLKS  000016
AS.WRT= 000002          D$OUTT  000043          LD$REL= 000004          N$FLG   000005          P.BUSY  000024
A$$CHK= 000000          D$RETF  000050          LD$RSV= 040000          N$FNC   000006          P.IOC   000003
A$$CPS= 000000          D$RNN   000002          LD$SUP= 000010          N$GENQ  000052          P.LNK   000000
A$$PRI= 000000          D$RTMR  000076          LNKS    000004R  002    N$GTM   000015          P.MAIN  000012
A$$TRP= 000000          D$SEG   000036          L$BASG  001000          N$HIGH  000033          P.NAM   000004
A.IOC   000003          D$SER   000032          L$BBLK  000360          N$LLT   000026          P.OWN   000026
A.LGTH= 000014          D$SQRL  000052          L$BDAT  000032          N$LLTM  000024          P.PRI   000002
A.MPCT  000011          D$$BUG= 177514          L$BEXT  000352          N$LVC   000036          P.PRO = ****** GX
A.PCB   000012          D$$ISK= 000000          L$BFLG  000030          N$MBXQ  000050          P.REL   000014
A.PCBL  000000          D$$L11= 000001          L$BFL2  000774          N$PLLT  000030          P.SIZE  000016
A.PRI   000002          D$$SYNC= 000000         L$BHGV  000012          N$SLA   000016          P.STAT  000030
A.STAT  000010          D$$SYNM= 000000         L$BHRB  000356          N$SNQD  000012          P.SUB   000010
A.TCB   000004          ERROR   000010R  002    L$BLDZ  000016          N$TIM   000004          P.SWSZ  000022
A.TCBL  000006          ERR10   000012          L$BLIB  000040          N$VCB   000010          P.TCB   000026
BLOX    000000R  002    ERR12   000014          L$BLRL  000776          N$$ACC= 000001          P.WAIT  000020
BUFUMP= 172354          ERR14   000016          L$BLUN  000362          N$$BUF= 000001          Q$$QPT= 000010
BYTES   000002R  002    ERR16   000020          L$BMXV  000014          N$$LDV= 000001          R$LDAT  000026
CMODE = 140000          ERR18   000022          L$BMXZ  000020          N$$MCP= 000001          R$LFLG  000024
C$LEN   000012          ERR2    000002          L$BOFF  000022          N$$MLL= 000001          R$LHGV  000006
C$NID   000000          ERR20   000024          L$BPAR  000004          N$$MOV= 000010          R$LLDZ  000012
C$NXH   000002          ERR22   000026          L$BPRI  000346          N$$NCT= 000001          R$LMXV  000010
C$TIM   000010          ERR24   000030          L$BROB  000364          N$$PEM= 000001          R$LMXZ  000014
C$$CKP= 000000          ERR26   000032          L$BROL  000366          PARHD = ****** GX       R$LNAM  000000
C$$ORE= 000400          ERR28   000034          L$BSA   000010          PCBA    000012R  002    R$LOFF  000016
C$$RSH= 177564          ERR30   000036          L$BSEG  000026          PIRQ  = 177772          R$LSA   000004
DEACB = ****** GX       ERR32   000040          L$BSGL  000354          PLGTH = ****** GX       R$LSEG  000022
DECPT = ****** GX       ERR34   000042          L$BSYS  000025          PMODE = 030000          R$LSIZ  000034
DSKBUF  000000R  003    ERR36   000044          L$BTSK  000000          PR.LBL  000156R         R$LWND  000020
D$AMXC  000072          ERR38   000046          L$BWND  000024          PR.PAR  000752R         R$$DER= 000000
D$AMXH  000074          ERR4    000004          L$BXFR  000350          PR.SUB  001332R         R$$K11= 000001
D$ANN   000000          ERR6    000006          L$$ASG= 000000          PR.XFR  001620R         R$$SND= 000000
D$BRPR  000102          ERR8    000010          L$$DRV= 000000          PR0   = 000000          R$$11M= 000000
D$BRTM  000100          EXTEND  000006R  002    L$$P11= 000001          PR1   = 000040          SWR   = 177570
D$DELF  000045          E$$XPR= 000000          L$$11R= 000000          PR2   = 000100          S$$WRG= 000000
D$DELW  000046          FE.EXV= ****** GX       MPAR  = 172100          PR3   = 000140          S$$YSZ= 007600
D$END = 000104          FE.PLA= ****** GX       MPCSR = 177746          PR4   = 000200          TPS   = 177564
D$FNB   000034          FL.CEX= ****** GX       M$$CRB= 000124          PR5   = 000240          TS$ACP= 020000
D$HIOR  000024          FL.LLC= ****** GX       M$$CRX= 000000          PR6   = 000300          TS$CHK= 000100
D$HOST  000022          FL.IOP= ****** GX       M$$FCS= 000000          PR7   = 000340          TS$CMP= 000200
D$INAC  000044          FMASK = ****** GX       M$$MGE= 000000          PS    = 177776          TS$IOP= 000020
D$INCT  000042          F$$LVL= 000001          M$$NET= 000000          PS.APR= 000007          TS$NEW= 000001
D$IPL   000051          G$$TPP= 000000          M$$OVR= 000000          PS.CHK= 010000          TS$NHD= 040000
D$LID   000020          G$$USS= 000000          NF$BLK= 000100          PS.CKP= 040000          TS$NSD= 002000
D$LNAM  000006          G$TTK= 000000           NF$DMO= 000010          PS.CKR= 020000          TS$NXH= 000002
D$LNUM  000014          G$$WRD= 000000          NF$MOU= 000040          PS.COM= 000200          TS$PIC= 100000
D$LST   000047          I$$RAR= 000000          NF$RST= 000002          PS.DEL= 000010          TS$PMD= 010000
D$MAXC  000064          I$$RDN= 000000          NF$SCN= 000020          PS.DRV= 000020          TS$PRV= 000400
D$MAXH  000066          KISAR0= 172340          NF$SHU= 000004          PS.FXD= 004000          TS$RES= 000040
D$MAXV  000070          KISAR6= 172354          NF$TIM= 000200          PS.LIO= 001000          TS$SLV= 004000
D$MLL   000040          K$$CNT= 177546          N$ACQ   000000          PS.NSF= 000400          TS$SUP= 000010
D$MNOD  000041          K$$CSR= 177546          N$ACTL  000032          PS.OUT= 100000          TS$XHR= 000004
D$NA    000062          K$$LDC= 000000          N$CIR   000034          PS.PER= 002000          T$$KMG= 000000
D$NBEA  000056          K$$TPS= 000074          N$DLA   000020          PS.PIC= 000100          T$$MIN= 000000
```

PROAL - PROCESS ALLOCATION ROUT MACRO V05.03b Saturday 29-Jun-85 00:37 Page 7-2
Symbol table

```
A$$CHK= 000000          E$$XPR= 000000          LD$LP = 000000          N$$BUF= 000001          R$$SND= 000000
A$$CPS= 000000          F$$LVL= 000001          L$$ASG= 000000          N$$LDV= 000001          R$$11M= 000000
A$$PRI= 000000          G$$TPP= 000000          L$$DRV= 000000          N$$MCP= 000001          S$$WRG= 000000
A$$TRP= 000000          G$$TSS= 000000          L$$P11= 000001          N$$MLL= 000001          S$$YSZ= 007600
C$$CKP= 000000          G$$TTK= 000000          L$$11R= 000000          N$$MOV= 000010          T$$KMG= 000000
C$$ORE= 000400          G$$WRD= 000000          M$$CRB= 000124          N$$NCT= 000001          T$$MIN= 000000
C$$RSH= 177564          I$$RAR= 000000          M$$CRX= 000000          N$$PEM= 000001          V$$CTR= 001000
D$$BUG= 177514          I$$RDN= 000000          M$$FCS= 000000          P$$P45= 000000          x$$DBT= 000000
D$$ISK= 000000          K$$CNT= 177546          M$$MGE= 000000          P$$WRD= 000000          $ALOC2  000006RG
D$$L11= 000001          K$$CSR= 177546          M$$NET= 000000          Q$$OPT= 000010          $DEAC2  000076RG
D$$YNC= 000000          K$$LDC= 000000          M$$OVR= 000000          R$$DER= 000000          $PRAVL  000002RG
D$$YNM= 000000          K$$TPS= 000074          N$$ACC= 000001          R$$K11= 000001

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000240    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8920 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.05
SY:PROAL.v2,[132,134]PROAL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]PROAL
```


PROAL CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 1

```
PROERR - PROCESS RELATED ERRORS MACRO V05.03b Saturday 29-Jun-85 00:37  Page 10-1
Symbol table

A$$CHK= 000000          FM.4A = 000000          M$$NET= 000000          T$$KMG= 000000          $ERR59  000662RG    002
A$$CPS= 000000          FM.5  = 000000          M$$OVR= 000000          T$$MIN= 000000          $ERR60  000710RG    002
A$$PRI= 000000          FM.5A = 000000          N$$ACC= 000001          V$$CTR= 001000          $EPR61  000746RG    002
A$$TRP= 000000          F$$LVL= 000001          N$$BUF= 000001          X$$DBT= 000000          $ER 62  000770RG    002
C$$CKP= 000000          G$$TPP= 000000          N$$LDV= 000001          $C5TA = ****** GX       $ERR63  001014RG    002
C$$ORE= 000400          G$$TSS= 000000          N$$MCP= 000001          $ERRQX  001506RG  002   $ERR64  001032RG    002
C$$RSH= 177564          G$$TTK= 000000          N$$MLL= 000001          $ERRQY  001544RG  002   $ERR65  001064RG    002
D$$BUG= 177514          G$$WRD= 000000          N$$MOV= 000010          $ERRXQ  001446RG  002   $ERR66  001130RG    002
D$$ISK= 000000          I$$RAR= 000000          N$$NCT= 000001          $ERRZA  001402RG  002   $ERR67  001160RG    002
D$$L11= 000001          I$$RDN= 000000          N$$PEM= 000001          $ERR44  000000RG  002   $ERR68  001216RG    002
D$$YNC= 000000          K$$CNT= 177546          PERR    000000R         $ERR45  000036RG  002   $ERR69  001260RG    002
D$$YNM= 000000          K$$CSR= 177546          P$$P45= 000000          $ERR46  000074RG  002   $ERR70  001332RG    002
ERRBUF  002272R   002  K$$LDC= 000000          P$$WRD= 000000          $ERR47  000122RG  002   $ERR9C  001700RG    002
ERRMSG  002262R   002  K$$TPS= 000074          Q$$OPT= 000010          $ERR48  000150RG  002   $ERR9D  001726RG    002
E$$XPR= 000000          LD$LP = 000000          REP4A   000066R         $ERR49  000234RG  002   $ERR9E  002006RG    002
FMT0    002064R   002  L$$ASG= 000000          REP5A   000154R         $ERR50  000256RG  002   $ERR99  001622RG    002
FMT4    002065R   002  L$$DRV= 000000          RTSPC   000064R         $ERR51  000322RG  002   $FNOUT= ****** GX
FMT4A   002115R   002  L$$P1?= 000001          R$$DER= 000000          $ERR53  000360RG  002   $NAME = ****** GX
FMT5    002156R   002  L$$11R= 000000          R$$K11= 000001          $ERR54  000410RG  002   $PNAME= ****** GX
FMT5A   002201R   002  M$$CRB= 000124          R$$SND= 000000          $ERR55  000462RG  002   $PRV.N= ****** GX
FMT5B   002240R   002  M$$CRX= 000000          R$$11M= 000000          $ERR56  000524RG  002   $REP.P= ****** GX
FMT9    002243R   002  M$$FCS= 000000          S$$WRG= 000000          $ERR57  000560RG  002   $SAVAL= ****** GX
FM.0  = 000000          M$$MGE= 000000          S$$YSZ= 007600          $ERR58  000616RG  002   $TIOUT= ****** GX
FM.4  = 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000232    001   (RW,I,LCL,REL,CON)
DATA    002400    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10137 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:23.01
SY:PROERR.?2,[132,134]PROERR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]PROERR
```



```
PROERR     CREATED BY  MACRO  ON 29-JUN-85 AT 00:37     PAGE 1
```

```
PROXPT
$ROL
```

$ROL - ALLOCATE REACHIBILITY MATRIX

```
                                                MOV     N$ROA1+2(R2),R0         ;;SET APR BIAS
                                                BEQ     45$                     ;;IF ENDNODE - BRANCH
                                                MOV     R2,R3                   ;;SAVE XPT DDB ADDRESS
                                                CALL    $MAPX                   ;;MAP TO ROL
                                                CLR     R0                      ;;SET UP VIRTUAL ADDRESS
                                                BIS     R2,R0                   ;; ...

                                                MOV     N$ROA1(R3),R1           ;;GET NODE ROL SIZE
                                                ADD     N$ROA2(R3),R1           ;;INCLUDE AREA ROL

                                        40$:    CLR     (R0)+                   ;;CLEAR ROL TABLE
                                                DEC     R1                      ;;COUNT ONE LESS ENTRY
                                                BNE     40$                     ;;IF MORE - BRANCH

                                                MOV     R3,R2                   ;;RESTORE XPT DDB
                                        45$:    SAVMAP                          ;;SAVE MAPPING
                                                MOV     N$TNC(R2),R0            ;;GET BYTES IN TRANSPORT NODE COUNTERS
                                                MOV     N$TNC+2(R2),R1          ;;AND POINTER TO TRANSPORT NODE COUNTERS

                                                MAP     Z,DSP(R5)               ;;MAP TO PROCESS SPACE
                                                CMP     #140000,R1              ;;APR 6 MAPPING ?
                                                BHI     50$                     ;;IF NO BRANCH

                                                ADD     #200,@KSAR5             ;;ACCESS NEXT 4K BLOCK OF XPT
                                        50$:    BIAS5   R1                      ;;BIAS R1 FOR APR5

                                        60$:    CLRB    (R1)+                   ;;CLEAR TRANSPORT NODE COUNTERS
                                                CMP     #140000,R1              ;;ACCESS NEXT BLOCK ?
                                                BNE     70$                     ;;IF NO - BRANCH

                                                MOV     #120000,R1              ;;ACCESS FIRST WORD OF NEXT BYTE
                                                ADD     #200,@KSAR5             ;;MAP TO NEXT 4K BLOCK

                                        70$:    SOB     R0,60$                  ;; ...

                                                RESMAP                          ;;RESTORE MAPPING
                                                RETURN

                                        80$:    MOV     R4,R0                   ;;GET # OF XPT CHANNELS
                                                DEC     R0                      ;;DON'T INCLUDE CHANNEL 0
                                                MOV     R0,N$TLC(R2)            ;;SAVE IN DATA DESCRIPTOR

                                                RETURN
```


$SKT - ALLOCATE CIRCUIT COUNTERS

```
                                ;+
                                ; CHECK - CHECK SIB RECORD FOR LEGAL FORMAT
                                ;
                                ; INPUTS:
                                ;       R3 = RECORD LENGTH
                                ;       R4 = RECORD ADDRESS
                                ;       R5 = POINTER TO FIRST SYMBOL BLOCK
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       V-BIT = SET IF NEITHER GSD NOR END GSD RECORD
                                ;       R3 INCREMENTED BY 2
                                ;       R4 DECREMENTED BY 2
                                ;       R5 = 0 IF END GSD RECORD
                                ;-
000326  005714                  CHECK:  TST     (R4)            ; FIRST WORD MUST BE BETWEEN 1 AND 6 INCLUSIVE
000330  001771                          BEQ     111$            ;
000332  022714  000006                  CMP     #6,(R4)         ;
000336  103766                          BLO     111$            ;
000340  162703  000002                  SUB     #2,R3           ; OMIT FIRST WORD FROM COUNT CONSIDERATIONS
000344  032703  000007                  BIT     #7,R3           ; COUNT MUST BE A MULTIPLE OF 8.
000350  001361                          BNE     111$            ;
000352  005324                          DEC     (R4)+           ; GSD RECORD ? (CLEARS V-BIT)
000354  001404                          BEQ     10$             ; YES ...
000356  005344                          DEC     -(R4)           ; END GSD RECORD ?
000360  000262                          SEV                     ; ASSUME NOT END GSD RECORD
000362  001001                          BNE     10$             ; NO ...
000364  005005                          CLR     R5              ; INDICATE END GSD SEEN (CLEARS V-BIT)
000366  000241                  10$:    CLC                     ; INDICATE SUCCESS
000370                                  RETURN
                                ;
                                        .DSABL  LSB
```

```
                                ;+
                                ; $PECHA - ''.PECHA'' ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000256                          $PECHA::
000256                                  CALL    $VXLLC          ; MAKE SURE IT'S NOT AN LLC
000262                                  CALL    $RQ1            ; REQUEST BINARY FILE
000266  103405                         BCS     10$             ; IF CS, NO ACCESS
000270  112712  000000G                MOVB    #.PECHA,(R2)    ; STORE FUNCTION CODE
000274  062767  000002  000000G        ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
000302                          10$:    RETURN
```

```
                                        ;+
                                        ; $MOD - MODIFICATION ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       .PNUMH=HIGH ORDER NUMERIC VALUE
                                        ;       .PNUMB=LOW ORDER NUMERIC VALUE
                                        ;       F2.SUB SET IF NEGATIVE MODIFICATION
                                        ;
                                        ; OUTPUTS:
                                        ;       $DIFF=MODIFICATION VALUE
                                        ;       F2.SUB CLEARED
                                        ;-
001260                                  $MOD::
001260  016767  000000G 000000G         10$:    MOV     .PNUMB,$DIFF    ; GET NUMERIC VALUE
001266  005767  000000G                         TST     .PNUMH          ; SINGLE PRECISION NUMBER?
001272  001014                                  BNE     10$             ; IF NE, NO - ERROR
001274  105767  000001G                         TSTB    $DIFF+1         ; IS VALUE CONTAINED IN 8 BITS?
001300  001011                                  BNE     10$             ; IF NE, NO - ERROR
001302  032767  000000G 000000G                 BIT     #F2.SUB,$FLAG2  ; NEGATIVE MODIFICATION?
001310  001410                                  BEQ     20$             ; IF EQ, NO
001312  005467  000000G                         NEG     $DIFF           ; NEGATE THE VALUE
001316  105467  000000G                         NEGB    .PNUMB          ; ANY SIGNIFICANCE LOST?
001322  100403                                  BMI     20$             ; IF MI, NO
001324                                          EMSG$   19              ; PRINT TRUNCATION WARNING
001332  042767  000000G 000000G         20$:    BIC     #F2.SUB,$FLAG2  ; CLEAR NEGATIVE FLAG
001340                                          RETURN
```

```
                                10$:    CLR     Z.DSP(R4)       ; ZERO DISPATCH TABLE ADDRESS
                                        CLR     Z.DSP+2(R4)     ; ZERO RELOCATION BIAS
                                        BIT     #ZF.LLC,Z.FLG(R4) ; IS THIS AN LLC?
                                        BEQ     20$             ; IF EQ, NO
                                        MOV     Z.DAT(R4),R3    ; GET LINE TABLE ADDRESS
                                        BEQ     20$             ; IF EQ, NO LINE TABLE ($XPRO)
                                        CMP     #-1,R3          ; IS THIS AN EMPTY LLC LINE TABLE?
                                        BEQ     19$             ; IF EQ, YES
                                        CLR     ..PDV           ; DISCARD ANY PARTITION LINE TABLES

                                        CMP     #^REVL,Z.NAM(R4) ; IS THIS EVL ?
                                        BNE     12$             ; IF NO - BRANCH

                                        SWSTK$  810$
                                        CLR     @LGPDV          ;; NO EVENT LOGGING PROCESS PRESENT NOW
                                        CLRB    @LGSTT          ;; CLEAR LOGGING STATS
                                        MOV     @LGDDB,R0       ;; GET ADDRESS OF LOGGING DATABASE
                                        BEQ     11$             ;; IF EQ, NOT PRESENT
                                        MOV     E$TCB(R0),R1    ;; GET ADDRESS OF EVENT COLLECTOR'S TCB
                                        BEQ     11$             ;; IF EQ, NOT ACTIVE
                                        CLR     E$TCB(R0)       ;; FORCE COLLECTOR TO EXIT
                                        MOV     R1,R0           ;; COPY EVC TCB ADDRESS
                                        CALL    @EXRQN          ;; UNSTOP EVENT COLLECTOR

                                11$:    RETURN

                                810$:   MOV     @LGDDB,R0       ; GET EVL DDB ADDRESS
                                        BEQ     12$             ; BR IF NO EVENT LOGGING
                                        DECL$S                  ; MAKE SURE EVC GETS SCHEDULED
                                811$:   MRKT$S  #1,#1,#2        ; WAIT 1 SECOND FOR EVC... TO CLEAN UP
                                        WTSE$S  #1              ; ...
                                        TST     E$STAT(R0)      ; HAS EVC... GONE AWAY YET?
                                        BNE     811$            ; BR IF NO
                                12$:    CMP     #^RXPT,Z.NAM(R4) ; IS THIS XPT?
                                        BNE     15$             ; BR IF NO
                                        CALL    $DEAHC          ; DEALLOCATE MINHOP/MINCOST,COUNTERS
                                        BR      18$             ; CONTINUE

                                15$:    CMP     #^RECL,Z.NAM(R4) ; IS THIS ECL?
                                        BNE     18$             ; BR IF NO
                                        CALL    DEAECL          ; DEALLOCATE ECL NODE COUNTERS
                                18$:    MOV     R4,-(SP)        ; SAVE PDV ADDRESS
                                        CALL    GETIND          ; GET PDV INDEX
                                        CALL    XLT16           ; DE-ALLOCATE LINE TABLE
                                        MOV     (SP)+,R4        ; RESTORE PDV ADDRESS
                                19$:    CLR     Z.DAT(R4)       ; ZERO LINE TABLE ADDRESS
                                20$:    SEC                     ; RETURN WITH C-BIT SET
                                        RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
                                101$:   CRASH                   ; NETLDR LOGIC ERROR
                                111$:   CRASH                   ; SOMEBODY UNLINKED OUR SUB-PCB!!!

                                        .DSABL  LSB
```

```
                                                CLR     R0              ;; INDICATE NO VECTORS FOUND
001524  005000                                  BR      48$             ;;
001526  000445
                                        30$:    MOV     R0,R4           ;; COPY VECTOR ADDRESS
001530  010004
                                                .IF DF  R$$MPL
                                                BIC     #160077,R4      ;; CLEAN OUT MAPPING BITS AND LOW 6 BITS
                                                .IFF
001532  042704  000077                          BIC     #77,R4          ;; DISCARD LOW 6 BITS
                                                .ENDC

001536                                          ASR$    5,R4            ;; CONVERT TO A WORD INDEX
001550  016404  000126'                         MOV     NSITAB(R4),R4   ;; GET APPROPRIATE VECTOR CONTENTS
001554  011404                                  MOV     (R4),R4         ;;  ...
001556  010005                                  MOV     R0,R5           ;; COPY VECTOR ADDRESS AGAIN
001560  042705  177700                         BIC     #^C<77>,R5      ;; GET LOW 6 BITS
001564  006205                                  ASR     R5              ;; DIVIDE BY 4
001566  006205                                  ASR     R5              ;;      ...
001570  062705  000000G                         ADD     #PR7,R5         ;; GIVING DEFAULT PRIORITY
001574                                          MTPS    #PR7            ;;; DISABLE INTERRUPTS
001602  010420                                  MOV     R4,(R0)+        ;;; RESTORE THE VECTOR
001604  010520                                  MOV     R5,(R0)+        ;;;  ...
001606                                          MTPS    #0              ;;; RE-ENABLE INTERRUPTS

001612  020001                                  CMP     R0,R1           ;; DONE?
001614  103004                                  BHIS    40$             ;; IF HIS, YES
001616  021002                                  CMP     (R0),R2         ;; DOES NEXT VECTOR POINT AT OUR LINE TABLE/ICB?
001620  103402                                  BLO     40$             ;; IF LO, NO
001622  021003                                  CMP     (R0),R3         ;; DITTO?
001624  103741                                  BLO     30$             ;; IF LO, YES
001626                                  40$:
                                        ;
                                        ; DE-ALLOCATE ICB(S)
                                        ;
                                                .IF DF  R$$MPL
                                                MOV     FMASK,FMKTMP    ;; GET $FMASK+2 ADDRESS
                                                ADD     #2,FMKTMP       ;;  ...
                                                BIT     #F2.DAS,@FMKTMP ;; IS DATA SPACE ENABLED ?
                                                BEQ     46$             ;; IF EQ, NO
                                                MOV     KRBA,R4         ;; GET THE KRB ADDRESS
                                                MOV     K.PRM(R4),R0    ;; GET ADDRESS OF 1ST ICB
                                                BEQ     44$             ;; IF EQ, NO ICB'S
                                                CLR     K.PRM(R4)       ;; CLEAR POINTER TO ICB CHAIN
                                                CLR     -(SP)           ;; ASSUME THE ICB IS NOT FOR A CPU PARTITION
                                                CMPB    #1,@NCPU        ;; ICB POINTER CONTAIN A CPU NUMBER ?
                                                BHIS    43$             ;; IF HIS, NO
                                                MOV     #120000,(SP)    ;; GET BIAS BITS FOR APR5 ADDRESSING
                                        41$:    MOV     R0,R2           ;; FORM CPU NUMBER IN UPPER FOUR BITS OF R2
                                                BIC     #7777,R2        ;;  ...
                                                BIC     R2,R0           ;; CLEAN CPU NUMBER OUT OF POINTER
                                                SWAB    R2              ;; FORM CPU NUMBER*2 OF POOL FOR THE ICB
                                                .REPT   3
                                                ASR     R2              ;;      ...
                                                .ENDR

                                                MOV     K6TAB,K6TMP     ;; GET $K6TAB ADDRESS
```

```
                        .SBTTL  XUMR - RELEASE UMR'S

                ;+
                ;
                ; XUMR - RELEASE UMR'S ON A MULTIPROCESSOR SYSTEM
                ;
                ; INPUTS:
                ;       R4 - NUMBER OF PROCESSES
                ;
                ; OUTPUTS:
                ;       UMR'S ARE RELEASED
                ;
                ;-

                        .IF DF  R$$MPL
XUMR:   SWSTK$  30$                     ;; ENTER KERNEL MODE
        MOV     @KSAR5,-(SP)            ;; SAVE CURRENT MAPPING
        ASL     R4                      ;; CONVERT NUMBER OF PROCESSORS TO WORD INDEX
10$:    SUB     #2,R4                   ;; DONE FOR ALL PROCESSORS ?
        BLT     20$                     ;; IF LT, YES
        MOV     K6TAB,K6TMP             ;; GET $K6TAB ADDRESS
        ADD     R4,K6TMP                ;; ...
        MOV     @K6TMP,@KSAR5           ;; MAP TO CPU PARTITION
        MOV     UMRPT,R0                ;; GET $UMRPT ADDRESS
        ADD     #120000,R0              ;; ...
        MOV     (R0),R0                 ;; GET POINTER TO UMR ASSIGNMENT BLOCK
        BEQ     10$                     ;; IF EQ, NONE .. KEEP LOOKING
        MOV     R0,R2                   ;; COPY UMR BLOCK ADDRESS
        CALL    DEUMR1                  ;; DE-ASSIGN MAPPING REGISTERS
        MOV     #M.LGTH,R1              ;; GET LENGTH OF UMR ASSIGNMENT BLOCK
        CALL    $DECEX                  ;; DE-ALLOCATE ASSIGNMENT BLOCK
        BR      10$                     ;; LOOK FOR MORE
20$:    MOV     (SP)+,@KSAR5            ;; RESTORE PREVIOUS MAPPING
30$:    RETURN                          ;; RETURN TO USER MODE AND TO CALLER
```

```
                                        .SBTTL  $DEAC2 - DEALLOCATE POOL SPACE TO AREA OUTSIDE TASK SPACE

                                ;+
                                ; *** - $DEAC2 - DEALLOCATE POOL SPACE TO AREA OUTSIDE TASK SPACE
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS OF THE CORE BUFFER TO BE DEALLOCATED.
                                ;       R1 = SIZE OF THE CORE BUFFER TO DEALLOCATE IN BYTES.
                                ;       R2 = DISPLACEMENT ADDRESS (100000 OR 120000).
                                ;       R3 = ADDRESS OF CORE ALLOCATION LISTHEAD-2.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       THE CORE BLOCK IS MERGED INTO THE FREE CORE CHAIN BY CORE
                                ;       ADDRESS AND IS AGLOMERATED IF NECESSARY WITH ADJACENT BLOCKS.
                                ;
                                ;-
004302                          DEAC2:
004302  061301                          ADD     (R3),R1         ;ROUND TO NEXT BOUNDARY
004304  042301                          BIC     (R3)+,R1        ;CLEAR EXCESS
004306  001442                          BEQ     60$             ;IF EQ NO BLOCK TO RELEASE
004310  010246                          MOV     R2,-(SP)        ;SAVE DISPLACEMENT
004312  010302                  20$:    MOV     R3,R2           ;SAVE ADDRESS OF CURRENT BLOCK
004314  011203                          MOV     (R2),R3         ;GET ADDRESS OF NEXT BLOCK
004316  001405                          BEQ     30$             ;IF EQ END OF CHAIN
004320  042703  160000                  BIC     #160000,R3      ;CLEAR CURRENT MAPPING REG.
004324  051603                          BIS     (SP),R3         ;USE OUR MAPPING REG.
004326  020003                          CMP     R0,R3           ;BLOCK GO HERE?
004330  103370                          BHIS    20$             ;IF HIS NO
004332  010310                  30$:    MOV     R3,(R0)         ;ASSUME NO AGLOMERATION
004334  010046                          MOV     R0,-(SP)        ;CALCULATE ADDRESS OF NEW BLOCK
004336  060116                          ADD     R1,(SP)         ;
004340  016646  000002                  MOV     2(SP),-(SP)     ;;CALCULATE END OF APR ADDRESSING
004344  052716  017777                  BIS     #17777,(SP)     ;;
004350  022616                          CMP     (SP)+,(SP)      ;;BIGGER THAN APR ADDRESSING RANGE?
004352  103423                          BLO     80$             ;;IF LO, YES
004354  020326                          CMP     R3,(SP)+        ;EQUAL TO NEXT IN CHAIN?
004356  103417                          BLO     70$             ;;IF LO, DEALLOCATION OVERLAPS
004360  001002                          BNE     40$             ;IF NE NO
004362  012310                          MOV     (R3)+,(R0)      ;MOVE LINK WORD TO BLOCK RELEASED
004364  061301                          ADD     (R3),R1         ;MERGE TWO BLOCKS
004366  010246                  40$:    MOV     R2,-(SP)        ;SAVE ADDRESS OF PREVIOUS BLOCK
004370  010012                          MOV     R0,(R2)         ;ASSUME NO AGLOMERATION
004372  061216                          ADD     (R2),(SP)       ;CALCULATE ADDRESS OF NEXT BLOCK
004374  020026                          CMP     R0,(SP)+        ;EQUAL TO BLOCK BEING RELEASED?
004376  001003                          BNE     50$             ;IF NE NO
004400  061201                          ADD     (R2),R1         ;MERGE TWO BLOCKS
004402  011042                          MOV     (R0),-(R2)      ;MOVE LINK WORD TO PREVIOUS BLOCK
004404  010200                          MOV     R2,R0           ;SET NEW ADDRESS OF BLOCK
004406  010160  000002          50$:    MOV     R1,2(R0)        ;SET SIZE OF BLOCK RELEASED
004412  012602                          MOV     (SP)+,R2        ;RESTORE ADDRESS OF LISTHEAD
004414                          60$:    RETURN

004416  005703                  70$:    TST     R3              ;;INSERT AT END OF LIST?
004420  001762                          BEQ     40$             ;;IF EQ YES, OK
004422                          80$:    CRASH                   ;;ILLEGAL DEALLOCATION
```



```
NTLUNL

SYMBOL

SYMBOL

LN.STA
LN.SUB
LN.TRI
L.COST
L.CTL
L.CVA
L.DDM
L.DDS
L.DLC
L.DLM
L.DLS
L.FLG
L.KRBA
L.LEN
L.MPF
L.NMST
L.NSTA
L.OWNR
L.UNT
MO$SAC
MO$SPR
MO$SSV
MO$25A
MO$25P
MO$25S
MO$29S
M$HIGH
M$$MGE

M$3100
M$3101
M$3102
M$3103
M.LGTH
NETPF
NSITAB
NS0
NS1
NS2
NS3
NS4
NS5
NS6
NS7
NTLPT
N$ENC
OBJHD
OP$INI
OP$TER
O.LEN
PDVID
PDVNM
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 6

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
LN.STA	= 000017	#5-72
LN.SUB	= 000360	#5-72
LN.TRI	= 000006	#5-72
L.COST	000015	#5-72
L.CTL	000012	#5-72 18-1033 18-1033
L.CVA	177776	#5-72
L.DDM	000002	#5-72 14-570 14-599 14-637 18-1031 18-1031 18-1047
L.DDS	000004	#5-72 *18-1036
L.DLC	000003	#5-72 11-443 14-637 14-642 14-646 18-1047
L.DLM	000006	#5-72 *11-446 *14-649
L.DLS	000010	#5-72 11-437 *11-445 14-639 *14-648 *18-1049
L.FLG	000000	#5-72 *14-568 *14-656 19-1095
L.KRBA	000016	#5-72 11-432 *11-435 14-576 14-625 15-680 *18-1037
L.LEN	= 000022	#5-72
L.MPF	000022	#5-72 11-427 18-1042
L.NMST	000020	#5-72
L.NSTA	000014	#5-72 11-424 18-1039
L.OWNR	000021	#5-72 *11-423 *18-1038
L.UNT	000013	#5-72
MO$SAC	= 000016	#5-66
MO$SPR	= 000012	#5-66
MO$SSV	= 000014	#5-66
MO$25A	= 000006	#5-66
MO$25P	= 000002	#5-66
MO$25S	= 000004	#5-66
MO$29S	= 000010	#5-66
M$HIGH	= 000003	#5-66
M$$MGE	= 000000	7-163 14-602 15-774 19-1179 19-1267 19-1318 27-1574 27-1621 27-1632 28-1720 28-1735 28-1762 29-1804
M$3100	= 000000	#5-66
M$3101	= 000001	#5-66
M$3102	= 000002	#5-66
M$3103	= 000003	#5-66 5-66
M.LGTH	= ****** GX	19-1255 19-1332
NETPF	= ****** GX	19-1076
NSITAB	000126 R	#7-179 15-793
NS0	= ****** GX	7-179
NS1	= ****** GX	7-180
NS2	= ****** GX	7-181
NS3	= ****** GX	7-182
NS4	= ****** GX	7-183
NS5	= ****** GX	7-184
NS6	= ****** GX	7-185
NS7	= ****** GX	7-186
NTLPT	= ****** GX	19-1194 19-1202 19-1269 27-1575 28-1721
N$ENC	000042	10-379 10-381 10-385
OBJHD	= ****** GX	23-1454
OP$INI	= 000000	#5-66
OP$TER	= 000001	#5-66
O.LEN	000012	23-1455
PDVID	= ****** GX	9-341
PDVNM	= ****** GX	19-1105

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 7

NTL11M
Symbol

U2.DH1=
U2.DJ1=
U2.DZ1=
U2.ESC=
U2.HFF=
U2.HLD=
U2.LOG=
U2.LWC=
U2.L3S=
U2.L8S=

. ABS.

Errors

*** Ass

Work f
Work f
Size of
Size of
Operati

Elapsed
SY:NTL1

NTL11M

NTL11M - RSX-11M DEFINITIONS MACRO V05.03b Saturday 29-Jun-85 00:35 Page 5-4
Symbol table

```
U2.DH1= 100000          U2.NEC= 004000          U3.UPC= 020000          $DEAC1= ****** G        $NS3  = ****** G
U2.DJ1= 040000          U2.PRV= 000010          U4.CR = 000100          $DEUMR= ****** G        $NS4  = ****** G
U2.DZ1= 000100          U2.RMT= 020000          V$$CTR= 001000          $DEVHD= ****** G        $NS5  = ****** G
U2.ESC= 001000          U2.R04= 100000          X$$SBT= 000000          $EXSIZ= ****** G        $NS6  = ****** G
U2.HFF= 010000          U2.SCS= 000004          $ALOCB= ****** G        $FMASK= ****** G        $NS7  = ****** G
U2.HLD= 000040          U2.SLV= 000200          $ALOC1= ****** G        $LBXL = 000340          $PARHD= ****** G
U2.LOG= 000400          U2.VT5= 000002          $ASUMR= ****** G        $NS0  = ****** G        $TKTCB= ****** G
U2.LWC= 000001          U2.7CH= 010000          $CLKHD= ****** G        $NS1  = ****** G        $$$   = 000062
U2.L3S= 000004          U3.OPA= 100000          $CRAVL= ****** G        $NS2  = ****** G        .TT0  = ****** G
U2.L8S= 010000          U3.PAR= 040000          $DEACB= ****** G

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 14662 Words  ( 58 Pages)
Size of core pool: 16552 Words  ( 63 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:11.07
SY:NTL11M.V2,[132,134]NTL11M/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTL11M
```


NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 1

```
000410  012605                          MOV     (SP)+,R5        ; RESTORE REGISTERS
000412  012604                          MOV     (SP)+,R4        ; ...
000414                                  RESMAP                  ; RESTORE APR 5 MAPPING
000420                          110$:   RESRG   <R3>            ; RESTORE R3
000422                                  RETURN                  ; AND RETURN

                                        .ENDC
```

```
                                        .SBTTL  $ALPRO - ALLOCATE STORAGE FOR PROCESS
                                ;+
                                ; *** - $ALPRO - TRY TO ALLOCATE STORAGE FOR PROCESS
                                ;
                                ;-

                                $ALPRO::
000106                                  CALL    $ALP16          ; TRY TO ALLOCATE STORAGE FROM PROCESS PARTITION
000106
000112  103002                          BCC     10$             ;IF CC, SUCCESS
000114                                  CALL    $ALL16          ;ELSE TRY TO ALLOCATE FROM CEX/RSX POOLS
000120                          10$:    RETURN

                                        .END
000001
```

```
                                        .SBTTL  PR.LBL - READ LABEL BLOCK

                                ;+
                                ; PR.LBL - READ LABEL BLOCK
                                ;
                                ; THIS ROUTINE IS CALLED TO READ THE PROCESS LABEL BLOCK AND TO VERIFY
                                ; THAT THE FILE IS A VALID PROCESS IMAGE. ON EXIT, THE LOGICAL BLOCK I/O
                                ; POINTERS ARE SETUP TO ACCESS THE FIRST TEXT BLOCK.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=LABEL BLOCK BUFFER ADDRESS
                                ;       R1,R2=DESTROYED
                                ;       $LBN=FIRST TEXT BLOCK LOGICAL BLOCK NUMBER
                                ;       $LLEN=PROCESS SIZE IN BYTES
                                ;       BLOX=PROCESS SIZE IN 32. WORD BLOCKS
                                ;-
000156  012700  000000'         PR.LBL: MOV     #DSKBUF,R0      ; GET BUFFER ADDRESS
000162                                  CALL    $RLBL           ; READ LABEL BLOCK
000166  103002                         BCC     1$              ; IF CC, SUCCESS
000170                                  CALLR   101$            ; IF CS - BRANCH
000174                          1$:
000174  032760  040000  000030          BIT     #TS$NHD,L$BFLG(R0) ; DOES PROCESS HAVE A HEADER?
000202  001002                         BNE     2$              ; IF NE, NO
000204                                  CALLR   111$            ; IF YES - ERROR

000210  016001  000016          2$:     MOV     L$BLDZ(R0),R1   ; GET LOAD SIZE IN BLOCKS
000214  010167  000020'                 MOV     R1,$LDSIZ       ; SAVE LOAD SIZE FOR LATER
000220  026001  000020                  CMP     L$BMXZ(R0),R1   ; IS PROCESS OVERLAID?
000224  001402                         BEQ     3$              ; IF EQ, NO
000226                                  CALLR   111$            ; IF NE, ERROR

000232                          3$:
                                        .IF DF  M$$MGE

000232  005002                         CLR     R2              ; ASSUME FOR COMM EXEC - NO EXTEND SIZE
000234  022767  114224  000000G        CMP     #^RXPT,$NAME    ; IS IT XPT ?
000242  001405                         BEQ     4$              ; IF YES - ANYTHING GOES

000244  022701  000200                 CMP     #200,R1         ; IS PROCESS > 4K WORDS?
000250  103002                         BCC     4$              ; IF CC, NO
000252                                  CALLR   131$            ; IF CS - ERROR

000256                          4$:
000256  032767  000000G 000000G        BIT     #FL.CEX,$FLAGS  ; IS THIS THE COMM EXEC?
000264  001142                         BNE     10$             ; IF NE, YES
000266  022760  120000  000010         CMP     #120000,L$BSA(R0) ; WAS PROCESS BUILT FOR APR5?
000274  001062                         BNE     6$              ; IF NE, NO

                                        .IFTF

000276  016002  000352                 MOV     L$BEXT(R0),R2   ; GET EXTEND SIZE
```

E 10

```
PRIO - READ PROCESS INTO CORE   MACRO V05.03b Saturday 29-Jun-85 00:36  Page 13-2
Symbol table

T2$CLI= 000001          X$$DBT= 000000          $LBXL = 000340          $PRCLN= ****** GX       $RLBL = ****** GX
T2$FMP= 000002          $CLDEQ= ****** GX       $LDSIZ  000020RG    002 $PREXT= ****** GX       $XPAR = ****** GX
UBMPR = 170200          $CLIOS= ****** GX       $LLEN = ****** GX       $PRINC= ****** GX       $$EXT = ****** GX
UISAR0= 177640          $CLOPE= ****** GX       $MUL  = ****** GX       $PRIO   000000RG        $$INC = ****** GX
UISAR1= 177642          $FLAGS= ****** GX       $NAME = ****** GX       $PRLLN= ****** GX       $$PAR = ****** GX
V$$CTR= 001000          $IOSB = ****** GX       $PNAME  000014RG    002 $READX= ****** GX       .BASEB= 140000


. ABS.  001000    000   (RW,I,GBL,ABS,OVR)
        001656    001   (RW,I,LCL,REL,CON)
DATA    000022    002   (RW,D,LCL,REL,CON)
..BUF   000002    003   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 13574 Words  ( 54 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:21.04
SY:PRIO.V2,[132,134]PRIO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]PRIO
```

F 10

```
PRIO        CREATED BY  MACRO  ON 29-JUN-85 AT 00:36     PAGE 1
```

PROAL CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
M$$MGE	= 000000	5-69
R$$11M	= 000000	5-61 6-99 7-160 7-185
$ALOC2	000006 RG	#6-100
$DEAC2	000076 RG	#7-145
$PRAVL	000002 RG	#5-64

PROAL CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 2

PROERR CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ERRBUF	002272 R	#6-150 7-170
ERRMSG	002262 R	#6-148 7-182
FMT0	002064 R	6-79 6-102 6-103 6-104 6-105 6-106 6-107 6-108 6-119
		6-120 6-121 6-122 #6-131
FMT4	002065 R	6-75 6-76 6-77 6-78 #6-132
FMT4A	002115 R	6-80 #6-133
FMT5	002156 R	6-81 6-82 6-83 6-84 6-85 6-87 6-88 6-89 6-101
		#6-134
FMT5A	002201 R	6-91 6-92 6-93 6-94 6-95 6-97 6-98 6-99 #6-135
FMT5B	002240 R	#6-136
FMT9	002243 R	#6-140
FM.0	= 000000	#6-79 #6-102 #6-103 #6-104 #6-105 #6-106 #6-107 #6-108 #6-119
		#6-120 #6-121 #6-122
FM.4	= 000000	#6-75 #6-76 #6-77 #6-78
FM.4A	= 000000	#6-80
FM.5	= 000000	#6-81 #6-82 #6-83 #6-84 #6-85 #6-87 #6-88 #6-89 #6-101
FM.5A	= 000000	#6-91 #6-92 #6-93 #6-94 #6-95 #6-97 #6-98 #6-99
I$$AS	= ******	6-109 6-137 10-266
PERR	000000 R	6-75 6-76 6-77 6-78 6-79 6-80 6-81 6-82 6-83
		6-84 6-85 6-87 6-88 6-89 6-91 6-92 6-93 6-94
		6-95 6-97 6-98 6-99 6-101 6-102 6-103 6-104 6-105
		6-106 6-107 6-108 6-119 6-120 6-121 6-122 #7-168
REP4A	000066 R	6-80 #8-205
REP5A	000154 R	6-91 6-92 6-93 6-94 6-95 6-97 6-98 6-99 #9-236
RTSPC	000064 R	6-79 6-102 6-103 6-104 6-105 6-106 6-107 6-108 6-119
		6-120 6-121 6-122 #7-190
R$$MPL	= ******	6-115
R$$11D	= ******	6-109 6-137 10-266
S$$BAS	= ******	6-75 6-75 6-76 6-76 6-77 6-77 6-78 6-78 6-79
		6-79 6-80 6-80 6-81 6-81 6-82 6-82 6-83 6-83
		6-84 6-84 6-85 6-85 6-87 6-87 6-88 6-88 6-89
		6-89 6-91 6-91 6-92 6-92 6-93 6-93 6-94 6-94
		6-95 6-95 6-97 6-97 6-98 6-98 6-99 6-99 6-101
		6-101 6-102 6-102 6-103 6-103 6-104 6-104 6-105 6-105
		6-106 6-106 6-107 6-107 6-108 6-108 6-119 6-119 6-120
		6-120 6-121 6-121 6-122 6-122
$C5TA	= ****** GX	8-213 9-243 9-246
$ERRQX	001506 RG	#6-107
$ERRQY	001544 RG	#6-108
$ERRXQ	001446 RG	#6-106
$ERRZA	001402 RG	#6-105
$ERR44	000000 RG	#6-75
$ERR45	000036 RG	#6-76
$ERR46	000074 RG	#6-77
$ERR47	000122 RG	#6-78
$ERR48	000150 RG	#6-79
$ERR49	000234 RG	#6-80
$ERR50	000256 RG	#6-81
$ERR51	000322 RG	#6-82
$ERR53	000360 RG	#6-83
$ERR54	000410 RG	#6-84
$ERR55	000462 RG	#6-85

PROERR CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 2

PROXPT
$CKT

31
31
31
31
31
31
31
31
32
32
32

```
                                        .SBTTL  $CKT    - ALLOCATE CIRCUIT COUNTERS
                                ;+
                                ; $CKT  - ALLOCATE CIRCUIT COUNTERS
                                ;
                                ; INPUTS:
                                ;       R2      - XPT DDB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       CIRCUIT COUNTERS ALLOCATED
                                ;-
                                ;
001062                          $CKT::  CALL    $SAVAL                  ;SAVE REGS
001066  016200  000100                  MOV     N$TLC(R2),R0            ;CHECK XPT CHANNELS
001072  001462                          BEQ     40$                     ;IF NONE - BRANCH
                                ;
001074  012701  000046                  MOV     #T$LLEN,R1              ;GET COUNTER LENGTH
001100                                  CALL    $MUL                    ;CALCULATE BLOCK NEEDED
001104  062701  000077                  ADD     #77,R1                  ;ROUND UP
001110  042701  000077                  BIC     #77,R1                  ;  ...
001114                                  ASR$    6,R1                    ;CONVERT TO BLOCKS
001130                                  CALL    $ALN18                  ;ALLOCATE BLOCK FROM NON-UMR POOL
001134  103004                          BCC     10$                     ;IF SUCCESS - BRANCH
001136                                  EMSG$   9D                      ;PRINT ERROR
001144  000435                          BR      40$                     ;CONTINUE
                                ;
001146  010062  000102          10$:    MOV     R0,N$TLC+2(R2)          ;STORE APR BIAS
001152  005000                          CLR     R0                      ;STORE VIRTUAL ADDRESS
001154                                  BIAS    R0                      ;  ...
001164  010062  000104                  MOV     R0,N$TLC+4(R2)          ;SET VIRTUAL ADDRESS
001170  016200  000102                  MOV     N$TLC+2(R2),R0          ;GET CIRCUIT COUNTER APR BIAS
                                ;
001174                                  SWSTK$  40$                     ;;ENTER SYSTEM STATE
001200  010203                          MOV     R2,R3                   ;;SAVE XPT DDB ADDRESS
001202                                  CALL    $MAPX                   ;;MAP TO COUNTER BLOCK
001206  005000                          CLR     R0                      ;;  ...
001210  050200                          BIS     R2,R0                   ;;SET UP VITUAL ADDRESS
001212  016301  000100                  MOV     N$TLC(R3),R1            ;;GET NUMBER OF COUNTERS
                                ;
001216  012720  177777          20$:    MOV     #-1,(R0)+               ;;MARK LINE COUNTER BLOCK AS FREE
001222  012703  000022                  MOV     #<<T$LLEN/2>-1>,R3      ;;GET REMAINING SIZE (IN WORDS)
                                ;
001226  005020                  30$:    CLR     (R0)+                   ;;ZERO REMAINDER OF BLOCK
001230                                  SOB     R3,30$                  ;;  ...
                                ;
001234                                  SOB     R1,20$                  ;;INITIALISE ALL BLOCKS
                                ;
001240                          40$:    RETURN                          ;;LEAVE SYSTEM STATE
                                ;
                                ;
                                ;
000001                                  .END
```

```
                                        ;+
                                        ; LOOK - LOOK FOR SYMBOL WITHIN SYMBOL TABLE RECORD
                                        ;
                                        ; INPUTS:
                                        ;       R0 = SYMBOL BLOCK ADDRESS
                                        ;       R3 = RECORD LENGTH-2
                                        ;       R4 = RECORD ADDRESS+2
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2 = DESTROYED
                                        ;       SB.DEF,SB.VAL SET IF DEFINITION IS FOUND
                                        ;-
000372  005001                  LOOK:  CLR     R1              ; SET POINTER TO START OF RECORD
000374  062704  000005                 ADD     #5,R4           ; DISPLACEMENT TO ENTRY TYPE FIELD
000400  020103                 10$:   CMP     R1,R3           ; AT END OF RECORD YET ?
000402  001430                        BEQ     20$             ; YES ..
000404  010102                        MOV     R1,R2           ; COPY CURRENT POINTER
000406  062701  000010                ADD     #8.,R1          ; ADJUST POINTER BY ENTRY LENGTH
000412  060402                        ADD     R4,R2           ; GIVING ADDRESS WITHIN RECORD BUFFER
000414  122712  000004                CMPB    #4,(R2)         ; GLOBAL SYMBOL VALUE ?
000420  001367                        BNE     10$             ; NO ..
000422  132742  000004                BITB    #4,-(R2)        ; NORMAL DEFINITION ?
000426  001364                        BNE     10$             ; NO ..
000430  026042  000006                CMP     SB.SYM+2(R0),-(R2) ; SECOND HALF OF NAME MATCH ?
000434  001361                        BNE     10$             ; NO ..
000436  026042  000004                CMP     SB.SYM(R0),-(R2) ; FIRST HALF MATCH ?
000442  001356                        BNE     10$             ; NO ..
000444  005266  000014                INC     12.(SP)         ; INDICATE DEFINITION FOUND
000450  016660  000014  000002        MOV     12.(SP),SB.DEF(R0) ; MARK SYMBOL DEFINED
000456  016260  000006  000010        MOV     6(R2),SB.VAL(R0) ; SET SYMBOL VALUE
000464  162704  000005         20$:   SUB     #5,R4           ; RESTORE R4
000470                                 RETURN
                                       .END
        000001
```


Symbol table

```
234                                     ;+
235                                     ; $END1 - ".END" OPERAND 1 ACTION ROUTINE
236                                     ;
237                                     ; INPUTS:
238                                     ;       $VALUE=TEMPLATE VERIFICATION SIZE
239                                     ;       $SZPTR=POINTER TO $SIZE VALUE AT START OF TEMPLATE
240                                     ;
241                                     ; OUTPUTS:
242                                     ;       ERROR MESSAGE PRINTED IF SIZE DOESN'T MATCH
243                                     ;-
244 000304                              $END1::
245 000304  005767  000000G                     TST     $FLAG1          ; HAS ANY ERROR OCCURRED?
246 000310  100412                              BMI     10$             ; IF MI, YES - DON'T COMPARE
247 000312  016700  000000G                     MOV     $SIZE,R0        ; GET CURRENT SIZE
248 000316  167700  000000G                     SUB     @$SZPTR,R0      ; LESS VALUE AT START OF TEMPLATE
249 000322  026700  000000G                     CMP     $VALUE,R0       ; DO SIZES MATCH?
250 000326  001403                              BEQ     10$             ; IF EQ, YES
251 000330                                      EMSG$   14              ; LENGTH OPERAND NOT SAME AS $SIZE
252 000336                              10$:    RETURN
```

```
                                        ;+
                                        ; $NOMOD - ACTION ROUTINE FOR NO MODIFICATIONS
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       $DIFF=0
                                        ;-
001342                                  $NOMOD::
001342  005067  000000G                 	CLR     $DIFF           ; ZERO THE MODIFICATION VALUE
001346                                  	RETURN
```

```
320
321
```



```
83
83
83
84
84
84
84
84
84
84
84
84
84
85
85
85
85
85
85
85
85
85
85
86
86
86
86
86
86
86
86
86
87
87
87
87
87
87
87
87
87
88
88
88
88
88
88
88
88
88
88
```

```
                                        ADD     R2,K6TMP        ;; ...
                                        MOV     @K6TMP,@KSAR5   ;; MAP TO THE ICB POOL
                                43$:    MOV     (SP)+,R2        ;; GET BIAS BITS
                                        ASL     R0              ;; FORM VIRTUAL ADDRESS OF ICB
                                        MOV     (R0),-(SP)      ;; SAVE POINTER TO NEXT ICB
                                        TST     -(R0)           ;; INCLUDE ICB NUMBER AND SIZE FIELD
                                        MOV     ICBSZ,R1        ;; GET LENGTH OF AN ICB
                                        MOV     ICAVL,R3        ;; GET $ICAVL-2 ADDRESS
                                        SUB     #2,R3           ;; POINT AT POOL LISTHEAD
                                        CALL    DEAC2           ;; DE-ALLOCATE THE ICB
                                        MOV     (SP)+,R0        ;; IS THERE ANOTHER ICB ?
                                        BEQ     44$             ;; IF EQ, NO
                                        MOV     R2,-(SP)        ;; DOES POINTER CONTAIN A CPU NUMBER ?
                                        BNE     41$             ;; IF NE, YES
                                        CRASH                   ;**** SHOULDN'T EVER GET HERE ? ****;
                                                                ;; ONLY ONE ICB PER CONTROLLER !
                                44$:    MOV     PC,R0           ;; INDICATE SUCCESS
                                45$:    MOV     (SP)+,@KSAR5    ;; RESTORE PREVIOUS MAPPING

                                        BR      48$             ;; AND RETURN TO USER MODE

                                        .IFTF   ; DF    R$$MPL

001626  016701  000006'         46$:    MOV     ICBSZ,R1        ;; GET SIZE OF ICB BLOCK
001632  001403                          BEQ     48$             ;; IF EQ, NO ICB BLOCK
001634  010200                          MOV     R2,R0           ;; COPY ADDRESS OF ICB

                                        .IFT    ; DF    R$$MPL

                                        CMP     -(R0),-(R0)     ;; DON'T FORGET LINK POINTER AND COUNT FIELDS

                                        .ENDC   ; DF    R$$MPL

001636                                  CALL    $DECEX          ;; DE-ALLOCATE ICB BLOCK

001642  010066  000002          48$:    MOV     R0,2(SP)        ;; UPDATE USER MODE R0
001646                                  RETURN                  ;; BACK TO USER MODE

001650                          50$:    .IF DF  R$$MPL
                                        TSTB    @NCPU           ; IS THIS A MULTIPROCESSOR SYSTEM ?
                                        BEQ     55$             ; IF EQ, NO
                                        RMAF$S                  ; ELSE, REMOVE PROCESSOR AFFINITY
                                        .ENDC   ; DF R$$MPL

001650  005700                  55$:    TST     R0              ; WAS AT LEAST 1 VECTOR FOUND?
001652  001402                          BEQ     111$            ; IF EQ, NO
001654                          60$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
001656                          101$:   CRASH                   ; LINE TABLE NOT FOUND WITHIN UNLOAD DATA
001660                          111$:   CRASH                   ; NO VECTORS POINTED AT LINE TABLE!
```

```
                        .SBTTL  DEUMR1 - DEASSIGN UNIBUS MAPPING REGISTERS

                ;+
                ;
                ;  DEUMR1 - DEASSIGN UNIBUS MAPPING REGISTERS
                ;
                ; INPUTS:
                ;       R2 - UMR BLOCK ADDRESS
                ;
                ; OUTPUTS:
                ;       UMR'S ARE DEASSIGNED
                ;
                ;-

DEUMR1: TST     (R2)+           ;; SKIP OVER LINK WORD
        MOV     UMRHD,R3        ;; POINT AT ASSIGNMENT LISTHEAD+2
        ADD     #120002,R3      ;; ...
10$:    CMP     -(R3),R2        ;; IS IT THE NEXT ONE ?
        BEQ     20$             ;; IF EQ, YES
        MOV     (R3),R3         ;; POINT TO NEXT IN LIST
        BNE     10$             ;; IF NE, MORE IN LIST
        BR      30$             ;; ELSE MAPPING ASSIGNMENT BLOCK NOT IN LIST
20$:    MOV     -(R2),(R3)      ;; UNLINK THE BLOCK FROM LIST
30$:    RETURN                  ;; RETURN TO CALLER
        .ENDC
```

```
                                        .SBTTL  DEALST - DEALLOCATE LIST OF DATA BLOCKS

                                ;+
                                ;
                                ; DEALST - DEALLOCATE LIST OF DATA BLOCKS
                                ;
                                ; INPUTS:
                                ;       R1 - LENGTH OF BLOCK TO DEALLOCATE
                                ;       R3 - ADDRESS OF LISTHEAD
                                ;
                                ; OUTPUTS:
                                ;       ALL BLOCKS IN THE LIST ARE DEALLOCATED AND THE LISTHEAD IS ZEROED
                                ;       R0,R2 DESTROYED
                                ;
                                ;-
004424                          DEALST::SWSTK$  30$             ;; ENTER SYSTEM STATE
004430                                  SAVMAP
004434  011346                         MOV     (R3),-(SP)      ;; STACK FIRST BLOCK ADDRESS
004436  005013                         CLR     (R3)            ;; AND ZERO LISTHEAD

004440  012600                  10$:    MOV     (SP)+,R0        ;; GET THE NEXT BLOCK IN LIST
004442  001412                         BEQ     20$             ;; IF EQ, END OF LIST
004444  010046                         MOV     R0,-(SP)        ;; SET BLOCK ADDRESS
004446                                  CALL    $CEACX          ;; GET ACCESS TO BLOCK
004452  012602                         MOV     (SP)+,R2        ;; GET MAPPED ADDRESS OF BLOCK
004454  011246                         MOV     (R2),-(SP)      ;; GET THE NEXT BLOCK IN LIST

004456                                  SAVRG   <R1>            ;; SAVE LENGTH OF BLOCK
004460                                  CALL    $XDEAC          ;; DE-ALLOCATE THE BLOCK
004464                                  RESRG   <R1>            ;; RESTORE LENGTH OF BLOCK
004466  000764                         BR      10$             ;; AND GO AGAIN

004470                          20$:    RESMAP
004474                          30$:    RETURN
```


NTLUNL
SYMBOL
SYMBOL
PDVTA
PH$HDE
PH$LOC
PH$MTS
PH$UMP
PH$WCS
PR7
PS
P.REL
RDBLH
RE$ADC
RE$ADF
RE$ADR
RE$BLK
RE$CAF
RE$DAT
RE$DRP
RE$LDT
RE$LSN
RE$NML
RE$OPE
RE$OPR
RE$RCV
RE$SED
RE$SKW
RE$STA
RE$SUM
RE$SYN
RE$TME
RE$TMO
RE$TMR
RE$UPT
RE$URE
RE$VER
RE$VRQ
RF.LOO
RT$INI
RT$OFF
RT$ON
R$R0
R$R1
R$R2
R$R3
R$R4
R$R5
R$$EIS
R$$MPL
R$$11D
R$$11M
R.ADD

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 7

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
PDVTA	= ******	GX	13-523	19-1103							
PH$HDE	= 000004		#5-66								
PH$LOC	= 000002		#5-66								
PH$MTS	= 000003		#5-66								
PH$UMP	= 000000		#5-66								
PH$WCS	= 000001		#5-66								
PR7	= ******	GX	13-518	15-799	15-800	19-1079					
PS	= ******	GX	*13-518	*13-543	*15-800	*15-803	*19-1079	*19-1087			
P.REL	= ******	GX	29-1838								
RDBLH	= ******	GX	19-1141								
RE$ADC	= 000004		#5-66								
RE$ADF	= 000017		#5-66								
RE$ADR	= 000007		#5-66								
RE$BLK	= 000010		#5-66								
RE$CAF	= 000014		#5-66								
RE$DAT	= 000001		#5-66								
RE$DRP	= 000016		#5-66								
RE$LDT	= 000013		#5-66								
RE$LSN	= 000012		#5-66								
RE$NML	= 000001		#5-66								
RE$OPE	= 000004		#5-66								
RE$OPR	= 000000		#5-66								
RE$RCV	= 000001		#5-66								
RE$SED	= 000011		#5-66								
RE$SKW	= 000006		#5-66								
RE$STA	= 000002		#5-66								
RE$SUM	= 000003		#5-66								
RE$SYN	= 000000		#5-66								
RE$TME	= 000021		#5-66								
RE$TMO	= 000000		#5-66								
RE$TMR	= 000020		#5-66								
RE$UPT	= 000002		#5-66								
RE$URE	= 000003		#5-66								
RE$VER	= 000005		#5-66								
RE$VRQ	= 000015		#5-66								
RF.LOO	= 100000		#5-71								
RT$INI	= 000002		#5-66								
RT$OFF	= 000001		#5-66								
RT$ON	= 000000		#5-66								
R$R0	= 000002		#6-110								
R$R1	= 000004		#6-111								
R$R2	= 000006		#6-112								
R$R3	= 000010		#6-113								
R$R4	= 000012		#6-114								
R$R5	= 000014		#6-115								
R$$EIS	= ******		10-384	15-792							
R$$MPL	= ******		5-83	7-139	8-244	11-412	14-579	14-592	14-610	15-685	15-712
			15-786	15-815	15-875	16-912	17-962	17-1004	19-1132	19-1180	19-1234
			19-1243	19-1319	21-1395						
R$$11D	= ******		10-384	15-792	27-1645	28-1777					
R$$11M	= 000000		19-1178	19-1179	19-1266	27-1577	28-1722	28-1764	29-1806		
R.ADD	000010		#5-71								

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 8

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
R.ADDR	000012		#5-71								
R.ADDX	000014		#5-71								
R.ADDXX	000016		#5-71								
R.ADDXXX	000020		#5-71								
R.ADDXXXX	000022		#5-71								
R.ADDXXXXX	000024		#5-71								
R.ADDXXXXXX	000026		#5-71								
R.ADDXXXXXXX	000030		#5-71								
R.ADDXXXXXXXX	000032		#5-71								
R.ADDXXXXXXXXX	000034		#5-71								
R.ADDXXXXXXXXXX	000036		#5-71								
R.ADDXXXXXXXXXXX	000040		#5-71								
R.ADDXXXXXXXXXXXX	000042		#5-71								
R.ADDXXXXXXXXXXXXX	000044		#5-71								
R.ADDXXXXXXXXXXXXXX	000046		#5-71								
R.ADDXXXXXXXXXXXXXXX	000050		#5-71								
R.ADDXXXXXXXXXXXXXXXX	000052		#5-71								
R.ADDXXXXXXXXXXXXXXXXX	000054		#5-71								
R.ADDXXXXXXXXXXXXXXXXXX	000056		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXX	000060		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXX	000062		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXX	000064		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXX	000066		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXX	000070		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXX	000072		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXX	000074		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXX	000076		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXX	000100		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000102		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000104		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000106		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000110		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000112		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000114		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000116		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000120		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000122		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000124		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000126		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000130		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000132		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000134		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000136		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000140		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000142		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000144		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000146		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000150		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000152		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000154		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000156		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000160		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000162		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000164		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000166		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000170		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000172		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000174		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000176		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000200		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000202		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000204		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000206		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000210		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000212		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000214		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000216		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000220		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000222		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000224		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000226		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000230		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000232		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000234		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000236		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000240		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000242		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000244		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000246		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000250		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000252		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000254		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000256		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000260		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000262		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000264		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000266		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000270		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000272		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000274		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000276		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000300		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000302		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000304		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000306		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000310		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000312		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000314		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000316		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000320		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000322		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000324		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000326		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000330		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000332		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000334		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000336		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000340		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000342		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000344		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000346		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000350		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000352		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000354		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000356		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000360		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000362		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000364		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000366		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000370		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000372		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000374		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000376		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000400		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000402		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000404		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000406		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000410		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000412		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000414		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000416		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000420		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000422		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000424		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000426		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000430		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000432		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000434		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000436		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000440		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000442		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000444		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000446		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000450		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000452		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000454		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000456		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000460		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	000462		#5-71								
R.ADDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES	
DV.MNT	= 100000	G	5-109	
D.DSP	000012	G	5-68	
D.NAM	000004	G	5-68	
D.PCB	000034	G	5-68	
D.UCB	000002	G	5-68	
D.UCBL	000010	G	5-68	
D.UNIT	000006	G	5-68	
D.VPWF	= 000006	G	5-68	
FE.CEX	= 020000	G	5-74	
FE.EXT	= 000001	G	5-74	
FE.EXV	= 000004	G	5-74	
FE.PLA	= 000020	G	5-74	
L$BEXT	000352	G	5-81	
L$BFLG	000030	G	5-81	
L$BHRB	000356	G	5-81	
L$BLDZ	000016	G	5-81	
L$BMXZ	000020	G	5-81	
L$BSA	000010	G	5-81	
M$$MGE	= 000000		5-117	
M.BFVH	000011	G	5-96	
M.BFVL	000012	G	5-96	
M.LGTH	= 000014	G	5-96	
M.UMRA	000002	G	5-96	
M.UMRN	000004	G	5-96	
PS.COM	= 000200	G	5-87	
PS.NSF	= 000400	G	5-87	
PS.SYS	= 000040	G	5-89	
P.ATT	= ******	G	5-117	
P.BLKS	000016	G	5-116	
P.BUSY	000024	G	5-89	
P.LGTH	= ******	G	5-116	
P.MAIN	000012	G	5-87	
P.NAM	000004	G	5-87	
P.PRO	= ******	G	5-117	
P.REL	000014	G	5-116	
P.SIZE	000016	G	5-116	
P.STAT	000030	G	5-87	
P.SUB	000010	G	5-87	
P.SWSZ	000022	G	5-116	
P.WAIT	000020	G	5-87	
R$$MPL	= ******		5-75	5-88
S.CON	000010	G	5-96	
S.LHD	000000	G	5-96	
TS$NHD	= 040000	G	5-81	
T.LDV	000044	G	5-102	
UC.PWF	= 000020	G	5-108	
US.BSY	= 000200	G	5-109	
US.MNT	= 000100	G	5-108	
U.CTL	000004	G	5-108	
U.CW1	000010	G	5-109	
U.DCB	000000	G	5-108	
U.RED	000002	G	5-108	

POLAL -
$LINKX -

300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339

NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

POLAL -

```
                                        .SBTTL  $LINKX - LINK/UNLINK BLOCK IN EXTENDED POOL
                                ;+
                                ; **-$LINKX-LINK/UNLINK BLOCK IN EXTENDED POOL
                                ;
                                ; THIS ROUTINE IS CALLED TO LINK OR UNLINK A BLOCK IN A LIST THAT MAY
                                ; INCLUDE ALLOCATIONS FROM EXTENDED POOL.
                                ;
                                ; INPUTS:
                                ;       TO LINK BLOCK INTO LIST:
                                ;          MOV X,-(SP) ; ADDRESS OF BLOCK TO BE LINKED INTO LIST
                                ;          MOV Y,-(SP) ; ADDRESS OF PREVIOUS BLOCK IN LIST
                                ;          CALL $LINKX
                                ;       TO UNLINK BLOCK FROM LIST:
                                ;          MOV X,-(SP) ; ADDRESS OF PREVIOUS BLOCK IN LIST
                                ;          MOV Y,-(SP) ; ADDRESS OF BLOCK TO BE UNLINKED FROM LIST
                                ;          CALL $LINKX
                                ;
                                ; OUTPUTS:
                                ;       BLOCK IS LINKED/UNLINKED, ADDRESSES ARE REMOVED FROM THE STACK.
                                ;
                                ; REGISTERS:
                                ;       NO REGISTERS ARE MODIFIED
                                ;-
                                $LINKX::
000424  010046                          MOV     R0,-(SP)                ; SAVE R0
000426  016646  000004                  MOV     4(SP),-(SP)             ; SET BLOCK ADDRESS FOR $CEACX
000432                                  CALL    $CEACX                  ; MAP TO BLOCK
000436  012600                          MOV     (SP)+,R0                ; GET MAPPED ADDRESS
000440  011046                          MOV     (R0),-(SP)              ; SAVE ADDRESS OF NEXT BLOCK IN LIST
000442  016610  000010                  MOV     10(SP),(R0)             ; SET NEW LINK WORD INTO BLOCK
000446  011046                          MOV     (R0),-(SP)              ; COPY ADDRESS OF BLOCK FOR $CEACX
000450                                  CALL    $CEACX                  ; MAP TO BLOCK
000454  012600                          MOV     (SP)+,R0                ; GET MAPPED ADDRESS
000456  012610                          MOV     (SP)+,(R0)              ; SET NEW LINK WORD
000460  012600                          MOV     (SP)+,R0                ; RESTORE R0
000462  012616                          MOV     (SP)+,(SP)              ; CLEAN OFF STACK (SAVING RETURN
000464  012616                          MOV     (SP)+,(SP)              ; ... ADDRESS)
000466                                  RETURN
```

```
PRALL - NTL PROCESS ALLOCATION  MACRO V05.03b Saturday 29-Jun-85 00:36  Page 7-1
Symbol table

A$$CHK= 000000          G$$TPP= 000000          L$$DRV= 000000          N$$MLL= 000001          T$$KMG= 000000
A$$CPS= 000000          G$$TSS= 000000          L$$P11= 000001          N$$MOV= 000010          T$$MIN= 000000
A$$PRI= 000000          G$$TTK= 000000          L$$11R= 000000          N$$NCT= 000001          V$$CTR= 001000
A$$TRP= 000000          G$$WRD= 000000          M$$CRB= 000124          N$$PEM= 000001          X$$DBT= 000000
C$$CKP= 000000          HEADR = ****** GX       M$$CRX= 000000          P$$P45= 000000          Z.AVL = ****** GX
C$$ORE= 000400          I$$RAR= 000000          M$$FCS= 000000          P$$WRD= 000000          $ALL16= ****** GX
C$$RSH= 177564          I$$RDN= 000000          M$$MGE= 000000          Q$$OPT= 000010          $ALOC2= ****** GX
D$$BUG= 177514          K$$CNT= 177546          M$$NET= 000000          R$$DER= 000000          $ALPRO  000106RG
D$$ISK= 000000          K$$CSR= 177546          M$$OVR= 000000          R$$K11= 000001          $ALP16  000000RG
D$$L11= 000001          K$$LDC= 000000          N$$ACC= 000001          R$$SND= 000000          $BIAS = ****** GX
D$$SYNC= 000000         K$$TPS= 000074          N$$BUF= 000001          R$$11M= 000000          $MAPX = ****** GX
D$$YNM= 000000          LD$LP = 000000          N$$LDV= 000001          S$$WRG= 000000          $PDVA = ****** GX
E$$XPR= 000000          L$$ASG= 000000          N$$MCP= 000001          S$$YSZ= 007600          $PRAVL= ****** GX
F$$LVL= 000001

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000122    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8941 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.38
SY:PRALL.V2,[132,134]PRALL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]PRALL
```



```
PRALL       CREATED BY  MACRO  ON 29-JUN-85 AT 00:36     PAGE 1
```

```
                                        .IFT
                                ;
                                ; CALCULATE LINE TABLES SIZE
                                ;
000302  005767  000002G                 IST     $PRINC+2        ; WAS THE LINE TABLE SIZE SPECIFIED ?
000306  001412                         BEQ     5$              ; IF EQ, NO .. ASSUME NO LINE TABLES
000310  010146                         MOV     R1,-(SP)        ; SAVE LOAD SIZE
000312  016701  000000G                MOV     $$INC,R1        ; GET THE LINE TABLE COUNT
000316  016700  000000G                MOV     $PRLLN,R0       ; GET THE LINE TABLE SIZE
000322                                 CALL    $MUL            ; GET SIZE IN BYTES OF LINE TABLES IN PROCESS
000326  010167  000006'                MOV     R1,EXTEND       ; SAVE THE SIZE OF THE LINE TABLES
000332  012601                         MOV     (SP)+,R1        ; RESTORE THE LOAD SIZE
000334  032767  000000G 000000G 5$:    BIT     #FL.LLC,$FLAGS  ; IS THIS AN LLC PROCESS ?
000342  001016                         BNE     7$              ; IF NE, YES
                                ;
                                ; DDM/DLC PROCESS
                                ;
000344  005767  000002G                 IST     $PREXT+2        ; WAS THE CONTROLLER SIZE GIVEN ?
000350  001473                         BEQ     9$              ; IF EQ, NO .. ASSUME NO CONTROLLERS
000352  010146                         MOV     R1,-(SP)        ; SAVE LOAD SIZE
000354  016701  000000G                MOV     $$EXT,R1        ; GET THE CONTROLLER COUNT
000360  016700  000000G                MOV     $PRCLN,R0       ; GET THE CONTROLLER TABLE SIZE
000364                                 CALL    $MUL            ; GET SIZE IN BYTES OF CONTROLLER TABLES
000370  060167  000006'                ADD     R1,EXTEND       ; ADD IN THE SIZE OF THE CONTROLLER TABLES
000374  012601                         MOV     (SP)+,R1        ; RESTORE THE LOAD SIZE
000376  000460                         BR      9$              ; AND SKIP LLC PROCESS EXTENSION
                                ;
                                ; LLC PROCESS
                                ;
000400                          7$:    .IFTF   ; DF M$$MGE

000400  016700  000000G                MOV     $$EXT,R0        ; GET THE EXTEND SIZE

                                        .IFF    ; DF M$$MGE

                                        BIT     #FL.LLC,$FLAGS  ; IS THIS AN LLC PROCESS ?
                                        BEQ     9$              ; IF EQ, NO
                                        ASL$    6,R0            ; CONVERT BLOCKS TO BYTES

                                        .ENDC   ; DF M$$MGE

000404  060002                         ADD     R0,R2           ; ADD IN THE PROCESS EXTENSION
000406  022767  114224  000000G         CMP     #^RXPI,$NAME    ; IS THIS XPI ?
000414  001066                         BNE     10$             ; IF NE, NO

000416                                 SWSTK$  12$             ;; ENTER SYSTEM STATE
000422  017701  000000G                MOV     @DECPT,R1       ;; GET DECNET HOME BLOCK ADDRESS
000426  016100  000060                 MOV     D$NN(R1),R0     ;; ENDNODE ?
000432  001404                         BEQ     15$             ;; IF YES - BRANCH
000434  005367  000004'                DEC     LNKS            ;; SET ROUTING FLAG
000440                                 RETURN

000442  000526                 6$:     BR      121$            ;; CONTINUE

000444  116167  000040  000004' 15$:   MOVB    D$MLL(R1),LNKS  ;; COPY MAX LOGICAL LINKS

000452                                 RETURN
```

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ALOCB	= ****** GX	11-599
BLOX	000000 R	#7-146 *9-414 *9-415 10-527 11-609 11-631 11-645
BUFUMP	= 172354	#5-74
BYTES	000002 R	#7-147 *9-420
CMODE	= 140000	#5-74
C$LEN	000012	9-386
DEACB	= ****** GX	11-667
DECPT	= ****** GX	9-367
DSKBUF	000000 R	#7-192 9-285 9-432
D$MLL	000040	9-375
D$NN	000060	9-368
ERROR	000010 R	#7-160 *8-222 8-235 8-244 *8-258 *8-260 *9-440 *9-442 *9-445
		*9-447 *9-449 10-553 *10-561 *10-563 *10-565 *10-567 *10-569 *10-572
		*11-662 *11-665 *13-793
ERR10	000012	#6-118 9-442
ERR12	000014	#6-119 9-445
ERR14	000016	#6-120 9-447
ERR16	000020	#6-121 9-449
ERR18	000022	#6-122 10-561
ERR2	000002	#6-114 8-258
ERR20	000024	#6-123 10-563
ERR22	000026	#6-124 10-565
ERR24	000030	#6-125 10-567
ERR26	000032	#6-126 10-569
ERR28	000034	#6-127 10-572
ERR30	000036	#6-128 11-662
ERR32	000040	#6-129 11-665
ERR34	000042	#6-130 13-793
ERR36	000044	#6-131
ERR38	000046	#6-132
ERR4	000004	#6-115
ERR6	000006	#6-116 8-260
ERR8	000010	#6-117 9-440
EXTE'ID	000006 R	#7-154 *9-331 *9-344 9-405
FE.EXV	= ****** GX	10-520
FE.PLA	= ****** GX	8-249
FL.CEX	= ****** GX	9-312 10-499 10-503
FL.LLC	= ****** GX	9-333
FL.TOP	= ****** GX	11-636 11-647
FMASK	= ****** GX	8-249 10-520
I$$AS	= ******	5-74 13-773
KISAR0	= 172340	#5-74
KISAR6	= 172354	#5-74
LNKS	000004 R	#7-148 *9-370 *9-375 9-381 9-385
L$BEXT	000352	9-319
L$BFLG	000030	9-290
L$BLDZ	000016	9-294
L$BMXZ	000020	9-296
L$BPAR	000004	10-478 10-479
L$BSA	000010	9-314 10-508
MPAR	= 172100	#5-74
MPCSR	= 177746	#5-74

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

PROAL CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES
CRASH	7-207
RETURN	6-124 7-203

FILEID**PROERR

PROERR CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
$ERR56	000524	RG	#6-87								
$ERR57	000560	RG	#6-88								
$ERR58	000616	RG	#6-89								
$ERR59	000662	RG	#6-91								
$ERR60	000710	RG	#6-92								
$ERR61	000746	RG	#6-93								
$ERR62	000770	RG	#6-94								
$ERR63	001014	RG	#6-95								
$ERR64	001032	RG	#6-97								
$ERR65	001064	RG	#6-98								
$ERR66	001130	RG	#6-99								
$ERR67	001160	RG	#6-101								
$ERR68	001216	RG	#6-102								
$ERR69	001260	RG	#6-103								
$ERR70	001332	RG	#6-104								
$ERR9C	001700	RG	#6-120								
$ERR9D	001726	RG	#6-121								
$ERR9E	002006	RG	#6-122								
$ERR99	001622	RG	#6-119								
$FNOUT	= ******	GX	6-75	6-76	6-77	6-78	6-82	6-83	6-84	6-85	6-87
			6-88	6-89	6-101						
$NAME	= ******	GX	8-212								
$PNAME	= ******	GX	9-242	9-244							
$PRV.N	= ******	GX	6-75	6-76	6-82	6-84	6-101				
$REP.P	= ******	GX	6-77	6-78	6-81	6-83	6-85	6-87	6-88	6-89	8-205
			9-236								
$SAVAL	= ******	GX	7-168								
$TIOUT	= ******	GX	7-185								

PROERR CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 3

PROXP
Symbol

AT$ACL
AT$ADN
AT$AUF
AT$CY(
AT$LV1
AT$LV2
AT$NEX
AT$UP
A$CIR
A$LEN
A$NID
A$TM
A$TMI
A$TSZ
A$TYP
A$$CH
A$$CP
A$$PR
A$$TRI
C$LEN
C$NID
C$NXH
C$TIM
C$$CK
C$$OR
C$$RS
DECPT
D$AMX
D$AMX
D$ANN
D$BRP
D$BRT
D$DEL
D$DEL
D$END
D$FNB
D$HIO
D$HOS
D$INA
D$INC
D$IPL
D$LID
D$LNA
D$LNU
D$LST
D$MAX

. ABS

Error

*** A

Work
Work
Size

PROXP

Symbol table

```
AT$ACL= 000100          D$MAXH  000066          I$$RAR= 000000          N$LVC   000036          T$LIFL  000013
AT$ADN= 000040          D$MAXV  000070          I$$RDN= 000000          N$LV1   000002          T$LIFO  000013
AT$AUP= 000020          D$MLL   000040          KSAR5 = ****** GX       N$LV2   000010          T$LIFS  000013
AT$CYC= 000004          D$MNOD  000041          K$$CNT= 177546          N$MBXQ  000050          T$LIN   000000
AT$LV1= 000002          D$NA    000062          K$$CSR= 177546          N$MHC1  000036          T$LIPS  000006
AT$LV2= 000001          D$NBEA  000056          K$$LDC= 000000          N$MHC2  000044          T$LLD   000012
AT$NEX= 000200          D$NBRA  000054          K$$TPS= 000074          N$PLD   000016          T$LLDC  000045
AT$UP = 000010          D$NEND= 000054          LD$LP = 000000          N$PLLT  000030          T$LLDL  000012
A$CIR   000003          D$NLN   000030          L$$ASG= 000000          N$PRI   000076          T$LLDO  000012
A$LEN   000004          D$NN    000060          L$$DRV= 000000          N$ROA1  000022          T$LLDS  000012
A$NID   000000          D$OUTT  000043          L$$P11= 000001          N$ROA2  000030          T$LLEN  000046
A$TM    000003          D$RETF  000050          L$$11R= 000000          N$RTMX  000014          T$LOPR  000002
A$TMI   000002          D$RNN   000002          M$$CRB= 000124          N$RTM1  000014          T$LTCL  000024
A$TSZ   000000          D$RTMR  000076          M$$CRX= 000000          N$RTM2  000015          T$LTIM  000026
A$TYP   000002          D$SEG   000036          M$$FCS= 000000          N$RT1   000000          T$_TPR  000014
A$$CHK= 000000          D$SER   000032          M$$MGE= 000000          N$RT2   000006          T$LTPS  000020
A$$CPS= 000000          D$SQRL  000052          M$$NET= 000000          N$SLA   000016          T$NAPL  000004
A$$PRI= 000000          D$$BUG= 177514          M$$OVR= 000000          N$SNOD  000012          T$NFE   000000
A$$TRP= 000000          D$$ISK= 000000          NF$BLK= 000100          N$TCTL  000112          T$NLEN  000010
C$LEN   000012          D$$L11= 000001          NF$DMO= 000010          N$TIM   000004          T$NNUL  000002
C$NID   000000          D$$YNC= 000000          NF$MOU= 000040          N$TLC   000100          T$NOPL  000006
C$NXH   000002          D$$YNM= 000000          NF$RST= 000002          N$TNC   000114          T$NRNI  000042
C$TIM   000010          E$NBR   000014          NF$SCN= 000020          N$TRC   000106          T$NRPL  000005
C$$CKP= 000000          E$NBS   000020          NF$SHU= 000004          N$TTCB  000110          T$NRUL  000007
C$$ORE= 000400          E$NCR   000034          NF$TIM= 000200          N$VCB   000010          T$NVR   000001
C$$RSH= 177564          E$NCS   000036          N$ACQ   000000          N$VER   000066          T$RPRI  000040
DECPT = ****** GX       E$NIC   000044          N$ACTL  000032          N$XLEN  000124          T$SVC   000034
D$AMXC  000072          E$NLEN  000050          N$ADJ1  000072          N$$ACC= 000001          T$T5    000030
D$AMXH  000074          E$NLLA  000012          N$ADJ2  000074          N$$BUF= 000001          T$T6    000032
D$ANN   000000          E$NLNK  000000          N$CACH  000062          N$$LDV= 000001          T$$KMG= 000000
D$BRPR  000102          E$NML   000040          N$CIR   000034          N$$MCP= 000001          T$$MIN= 000000
D$BRTM  000100          E$NMR   000024          N$CRC   000120          N$$MLL= 000001          V$$CTR= 001000
D$DELF  000045          E$NMS   000030          N$DLA   000020          N$$MOV= 000010          X$$DBT= 000000
D$DELW  000046          E$NNOD  000002          N$DLY   000014          N$$NCT= 000001          Z.DSP = ****** GX
D$END = 000104          E$NRT   000042          N$ELEN  000054          N$$PEM= 000001          Z.LLN = ****** GX
D$FNB   000034          E$NRTP  000005          N$ENC   000042          P$$P45= 000000          $ADJ    000172RG
D$HIOR  000024          E$NSEG  000010          N$ERRC  000022          P$$WRD= 000000          $ALN18= ****** GX
D$HOST  000022          E$NTIM  000046          N$FLG   000005          Q$$OPT= 000010          $CKT    001062RG
D$INAC  000044          E$NUSE  000004          N$FNC   000006          R$$DER= 000000          $ERR9D= ****** GX
D$INCT  000042          E$STRT  000006          N$GENQ  000052          R$$K11= 000001          $MAPX = ****** GX
D$IPL   000051          E$$XPR= 000000          N$GTM   000015          R$$SND= 000000          $MUL  = ****** GX
D$LID   000020          F$$LVL= 000001          N$HC1   000052          R$$11M= 000000          $NICAH  000000RG
D$LNAM  000006          G$$TPP= 000000          N$HC2   000056          S$$WRG= 000000          $PDVA = ****** GX
D$LNUM  000014          G$$TSS= 000000          N$HIGH  000033          S$$YSZ= 007600          $ROL    000466RG
D$LST   000047          G$$TTK= 000000          N$LLT   000026          T$FLAG  000044          $SAVAL= ****** GX
D$MAXC  000064          G$$WRD= 000000          N$LLTM  000024          T$LIF   000013

. ABS.  000124    000   (RW,I,GBL,ABS,OVR)
        001242    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file reads: 0
Work  file writes: 0
Size of work file: 14528 Words  ( 57 Pages)
```

```
SYMBL - SYMBOL TABLE FILE LOOKU MACRO V05.03b Saturday 29-Jun-85 00:38  Page 9-1
Symbol table

A$$CHK= 000000          E.ORUN  000006          LOOK    000372R         N$$NCT= 000001          S$$YSZ= 007600
A$$CPS= 000000          E.READ  000004          L$$ASG= 000000          N$$PEM= 000001          T$$KMG= 000000
A$$PRI= 000000          F$$LVL= 000001          L$$DRV= 000000          N.FTYP= ****** GX       T$$MIN= 000000
A$$TRP= 000000          F.DFNB= ****** GX       L$$P11= 000001          P$$P45= 000000          V$$CTR= 001000
CHECK   000326R         G$$TPP= 000000          L$$11R= 000000          P$$WRD= 000000          X$$DBT= 000000
CL.OPN= ****** GX       G$$TSS= 000000          M$$CRB= 000124          Q$$OPT= 000010          $CLBUF= ****** GX
C$$CKP= 000000          G$$TTK= 000000          M$$CRX= 000000          READ    000222R         $CLDEQ= ****** GX
C$$ORE= 000400          G$$WRD= 000000          M$$FCS= 000000          R$$DER= 000000          $CLFDB= ****** GX
C$$RSH= 177564          IE.EOF= ****** GX       M$$MGE= 000000          R$$K11= 000001          $CLFLG= ****** GX
D$$BUG= 177514          IE.RBG= ****** GX       M$$NET= 000000          R$$SND= 000000          $CLIOS= ****** GX
D$$ISK= 000000          I$$RAR= 000000          M$$OVR= 000000          R$$11M= 000000          $CLQUE= ****** GX
D$$L11= 000001          I$$RDN= 000000          N$$ACC= 000001          SB.DEF  000002          $CLSAV= ****** GX
D$$YNC= 000000          K$$CNT= 177546          N$$BUF= 000001          SB.LNK  000000          $CLSIZ= ****** GX
D$$YNM= 000000          K$$CSR= 177546          N$$LDV= 000001          SB.SYM  000004          $GCLEP= ****** GX
E$$XPR= 000000          K$$LDC= 000000          N$$MCP= 000001          SB.VAL  000010          $SAVAL= ****** GX
E.BAD   000010          K$$TPS= 000074          N$$MLL= 000001          STBBF   000000R     002 $$SYMBL  000000RG
E.OPEN  000002          LD$LP = 000000          N$$MOV= 000010          S$$WRG= 000000

. ABS.  000012    000   (RW,I,GBL,ABS,OVR)
        000472    001   (RW,I,LCL,REL,CON)
DATA    000174    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8951 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.57
SY:SYMBL.V2,[132,134]SYMBL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]SYMBL
```



```
SYMBL       CREATED BY  MACRO  ON 29-JUN-85 AT 00:38     PAGE 1
```

```
                                        ;+
                                        ; $END2 - ".END" OPERAND 2 AC"ION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=TEMPLATE POINTER OFFSET
                                        ;
                                        ; OUTPUTS:
                                        ;       $OFF=CURRENT COPY OF $VALUE
                                        ;-
000340                                  $END2::
000340  016767  000000G 000000G         10$:    MOV     $VALUE,$OFF     ; COPY OFFSET VALUE
000346                                          CALL    $VFLV1          ; MAKE SURE IT'S LEVEL 1 TEMPLATE
000352  032767  000000G 000000G                 BIT     #FL.MUX,$FLAGS  ; MUX LINE?
000360  001010                                  BNE     101$            ; IF NE, YES - OFFSET NOT ALLOWED
000362  005767  000000G                         TST     $FLAG1          ; HAS ANY ERROR OCCURRED?
000366  100404                                  BMI     10$             ; IF MI, YES - DON'T CHECK
000370  026767  000000G 000000G                 CMP     $VALUE,$SIZE    ; IS OFFSET WITHIN LINE TABLE?
000376  103004                                  BHIS    111$            ; IF HIS, NO
000400                                          RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000402                                  101$:   EMSG$R  2               ; OFFSET NOT ALLOWED FOR MUX LINES
000410                                  111$:   EMSG$R  15              ; OFFSET VALUE NOT WITHIN LINE TABLE
```

```
                                        ;+
                                        ; $FNAME - FILE NAME ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT=FILE NAME ADDRESS
                                        ;       .PSTCN=FILE NAME LENGTH
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $FILEN=ASCIZ FILE NAME
                                        ;       F2.FIL SET
                                        ;-
001350                                  $FNAME::
001350  016700  000000G                         MOV     .PSTPT,R0       ; GET SOURCE ADDRESS
001354  012701  000000G                         MOV     #$FILEN,R1      ; TARGET ADDRESS
001360  016702  000000G                         MOV     .PSTCN,R2       ; AND LENGTH
001364  022702  000011                          CMP     #9.,R2          ; 9 CHARACTERS OR LESS?
001370  103413                                  BLO     101$            ; IF LO, NO - ERROR
001372  112011                          10$:    MOVB    (R0)+,(R1)      ; COPY FILE NAME
001374  122721  000056                          CMPB    #'.,(R1)+       ; IS IT A PERIOD?
001400  001412                                  BEQ     111$            ; IF EQ, YES - ERROR
001402                                          SOB     R2,10$
001406  105011                                  CLRB    (R1)            ; MAKE IT AN ASCIZ STRING
001410  052767  000000G 000000G                 BIS     #F2.FIL,$FLAG2  ; INDICATE VALID FILE NAME SEEN
001416                                          RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
001420                                  101$:   EMSG$R  20              ; FILE NAME IS TOO LONG
001426                                  111$:   EMSG$R  52              ; FILE NAME CONTAINS A PERIOD
```

```
                                        .SBTTL  $DSINI  INIT/TERM PROCESS VIA DIRECT CALL
                                ;+
                                ;
                                ; $DSINI - CALL PROCESS AT INIT OR TERM DISPATCH ADDRESS
                                ;
                                ; INPUTS:
                                ;       R1      - PDV ADDRESS
                                ;       $$PDIN  - FUNCTION CODE
                                ;
                                ; OUTPUTS:
                                ;       PROCESS CALLED AT INIT/TERM DISPATCH
                                ;       C-BIT   - SUCCESS/FAILURE
                                ;-
000336                          $DSINI::SWSTK$  10$             ; ENTER SYSTEM STATE
000342  016102  000000G                 MOV     Z.NAM(R1),R2    ; COPY PROCESS NAME
000346  016105  000000G                 MOV     Z.DAT(R1),R5    ; COPY PROCESS DATA BASE ADDRESS
000352                                  CALL    @PDVID          ; GET PDV INDEX
0003[illegible]  016746  000000G        MOV     $$PDIN,-(SP)    ; GET ADDRESS OF FUNCTION CODE
000362  010603                          MOV     SP,R3           ; STORE ADDRESS OF FUNCTION CODE

000364                                  CALL    @LLCAL          ; DISPATCH TO PROCESS
000370  005726                          TST     (SP)+           ; CLEAN UP STACK

000372  010566  000002                  MOV     R5,2(SP)        ; LOAD DDB STATUS ADDRESS INTO USER R0

000376                                  RETURN

000400  032710  000001          10$:    BIT     #INIFL,(R0)     ; HAS PROCESS COMPLETED ?
000404  001004                          BNE     20$             ; IF YES - BRANCH
000406                                  WSIG$S                  ; IF NO - WAIT
000414  000771                          BR      10$             ; TRY AGAIN

000416  000241                  20$:    CLC                     ; SET SUCCESS STATUS

000420                                  RETURN
```

```
                                        .SBTTL  ICBSC - DETERMINE IF VECTOR POINTS AT ICB
                                ;+
                                ; *** - ICBSC - DETERMINE IF VECTOR POINTS AT ICB
                                ;
                                ; INPUT:
                                ;       R0 = VECTOR ADDRESS
                                ;       R1 = HIGH LIMIT FOR LEGAL VECTORS
                                ;       R2 = LINE TABLE START ADDRESS
                                ;       R3 = ADDRESS OF END OF LINE TABLE
                                ;       $BIAS = BIAS FOR THIS PROCESS
                                ;
                                ; OUTPUT:
                                ;       C-BIT   SUCCESS/FAILURE
                                ;               IF SUCCESS:
                                ;                               R2 = START ADDRESS OF ICB
                                ;                               R3 = END ADDRESS OF ICB
                                ;       R4,R5 = DESTROYED
                                ;
                                ;-
001662                          ICBSC:
                                        .IF DF  R$$MPL
                                        MOV     FMASK,FMKTMP    ;; GET $FMASK+2 ADDRESS
                                        ADD     #2,FMKTMP       ;; ...
                                        BIT     #F2.DAS,@FMKTMP ;; IS DATA SPACE ENABLED ?
                                        BNE     5$              ;; IF NE, YES
                                        .ENDC
001662  020227  120000                  CMP     R2,#120000      ;; PARTITION LINE TABLE?
001666  103412                         BLO     10$             ;; IF LO, NO - NOT AN ICB
001670  032710  000001          5$:     BIT     #1,(R0)         ;; ICB ON WORD BOUNDARY?
001674  001007                         BNE     10$             ;; IF NE, NO - NOT AN ICB
001676  011004                         MOV     (R0),R4         ;; GET ADDRESS OF ASSUMED ICB
001700                                  CALL    ICBLEN          ;; CALCULATE LENGTH IF THIS IS AN ICB
001704  103404                         BCS     20$             ;; IF CS, LEAVE
001706  011002                         MOV     (R0),R2         ;; GET START ADDRESS OF ICB IN R2
001710  060203                         ADD     R2,R3           ;; CALCULATE END ADDRESS OF ICB
001712  000401                         BR      20$             ;; AND LEAVE
001714  000261          10$:    SEC                             ;; INDICATE FAILURE
001716                  20$:    RETURN                          ;; RETURN TO CALLER
```

```
                                        .SBTTL  XOBJTB - DE-ALLLOCATE OBJECT TABLE
                                ;+
                                ; XOBJTB - DE-ALLOCATE THE OBJECT TABLE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ALL OBJECT BLOCKS ARE DE-ALLOCATED
                                ;-
003124  016703  000000G         XOBJTB: MOV     OBJHD,R3        ; POINT AT START OF OBJECT TABLE LIST
003130  012701  000012                  MOV     #O.LEN,R1       ; GET THE LENGTH OF AN OBJECT BLOCK
003134                                  CALL    DEALST          ; DEALLOCATE THE OBJECT BLOCK TABLE
003140                                  RETURN
```

```
                                        .SBTTL  XSERTB DEALLOCATE SERVICE BLOCKS
                                ;+
                                ; XSERTB - DEALLOCATE SERVICE BLOCKS
                                ;
                                ; INPUTS -
                                ;       NONE
                                ;
                                ; OUTPUTS -
                                ;       SERVICE BLOCKS DEALLOCATED
                                ;-
004476  012701  000052          XSERTB: MOV     #S.LENG,R1      ; GET BLOCK LENGTH
004502  017703  000000G                 MOV     @DECPT,R3       ; GET DECNET HOME BLOCK ADDRESS
004506  001404                         BEQ     10$             ; IF PSI ONLY SYSTEM - BRANCH
004510  062703  000032                 ADD     #D$SER,R3       ; GET SERVICE BLOCK LISTHEAD

004514                                  CALL    DEALST          ; DEALLOCATE ALL BLOCKS

004520                          10$:    RETURN



                                ;+
                                ; XFILTB - DEALLOCATE FILENAME BLOCK
                                ;
                                ; INPUTS -
                                ;       NONE
                                ;
                                ; OUTPUTS -
                                ;       ALL FILENAME BLOCKS DEALLOCATED
                                ;-

004522  017703  000000G         XFILTB: MOV     @DECPT,R3       ; GET DECNET HOME BLOCK ADDRESS
004526  001436                         BEQ     30$             ; IF PSI ONLY SYSTEM - BRANCH
004530  062703  000034                 ADD     #D$FNB,R3       ; GET FILENAME BLOCK LISTHEAD

004534                                  SWSTK$  30$             ; ENTER SYSTEM STATE
004540                                  SAVMAP                  ; SAVE MAPPING

004544  011300                  10$:    MOV     (R3),R0         ; GET NEXT FILENAME BLOCK ADDRESS
004546  001424                         BEQ     20$             ; IF END OF LIST - BRANCH

004550  010046                         MOV     R0,-(SP)        ; SET UNMAPPED ADDRESS
004552                                  CALL    $CEACX          ; GET ACCESS TO BLOCK
004556  012604                         MOV     (SP)+,R4        ; GET MAPPED ADDRESS
004560  011413                         MOV     (R4),(R3)       ; SET NEW LISTHEAD ADDRESS

004562  010401                         MOV     R4,R1           ; COPY MAPPED ADDRESS
004564  005002                         CLR     R2              ; INIT REG
004566  062701  000003                 ADD     #F.USE+1,R1     ; CALCULATE SIZE OF BLOCK
004572  111102                         MOVB    (R1),R2         ; GET LENGTH OF FIRST FILED
004574  060201                         ADD     R2,R1           ; GET ADDRESS OF NEXT FIELD LENGTH
004576  005046                         CLR     -(SP)           ; INIT STACK LOCATION

004600  111116                         MOVB    (R1),(SP)       ; GET NEXT FIELD LENGTH
004602  061601                         ADD     (SP),R1         ; GET ADDRESS OF NEXT FIELD LENGTH
004604  111116                         MOVB    (R1),(SP)       ; GET NEXT FIELD LENGTH
```



```
NTLUNL
SYMBOL
SYMBOL
R.FLAG
R.LEN
R.LNK
R.NAM
SCAN
SC$OFF
SC$ON
SC$RST
SC$SHU
SDBLH
SF.ALT
SF.CIR
SF.DEV
SF.DIA
SF.DPA
SF.DPC
SF.DUM
SF.ENA
SF.HAD
SF.HST
SF.LOA
SF.LPB
SF.MFL
SF.PAC
SF.PH3
SF.PSS
SF.REA
SF.SEC
SF.SER
SF.SVC
SF.TER
SF.UNL
SLTMA
SLTNM
SV$DUM
SV$LOA
S.ADD
S.CIR
S.COST
S.DEV
S.DIA
S.DPA
S.DPC
S.DUM
S.FLAG
S.FLG
S.HAD
S.HST
S.LEN
S.LENG
S.LNK
S.LOA
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 8

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
R.FLAG	000012		#5-71		
R.LEN	000014		#5-71	25-1518	
R.LNK	000000		#5-71		
R.NAM	000002		#5-71		
SCAN	= ******	GX	14-569		
SC$OFF	= 000001		#5-66		
SC$ON	= 000000		#5-66		
SC$RST	= 000003		#5-66		
SC$SHU	= 000002		#5-66		
SDBLH	= ******	GX	19-1144		
SF.ALT	= 000200		#5-72		
SF.CIR	= 000010		#5-74		
SF.DEV	= 000020		#5-74		
SF.DIA	= 004000		#5-74		
SF.DPA	= 000100		#5-74		
SF.DPC	= 000200		#5-74		
SF.DUM	= 010000		#5-74		
SF.ENA	= 000100		#5-72		
SF.HAD	= 000004		#5-74		
SF.HST	= 000002		#5-74		
SF.LOA	= 002000		#5-74		
SF.LPB	= 000004		#5-72		
SF.MFL	= 000040		#5-72		
SF.PAC	= 000020		#5-72		
SF.PH3	= 020000		#5-74		
SF.PSS	= 000040		#5-74		
SF.REA	= 000010		#5-72		
SF.SEC	= 000400		#5-74		
SF.SER	= 000001		#5-72		
SF.SVC	= 000002		#5-72		
SF.TER	= 001000		#5-74		
SF.UNL	= 000040		#5-72		
SLTMA	= ******	GX	18-1027	19-1090	
SLTNM	= ******	GX	18-1029	19-1091	
SV$DUM	= 000001		#5-66		
SV$LOA	= 000000		#5-66		
S.ADD	000004		#5-74		
S.CIR	000016		#5-74		
S.COST	000001		#5-72		
S.DEV	000020		#5-74		
S.DIA	000046		#5-74		
S.DPA	000032		#5-74		
S.DPC	000036		#5-74		
S.DUM	000050		#5-74		
S.FLAG	000002		#5-74		
S.FLG	000000		#5-72		
S.HAD	000010		#5-74		
S.HST	000006		#5-74		
S.LEN	000004		#5-72	11-429	18-1044
S.LENG	000052		#5-74	32-1999	
S.LNK	000000		#5-74		
S.LOA	000044		#5-74		

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 9

NTL11M

SYMBOL

SYMBOL

U.SCB
U.STS
U.UNIT
$ALOCB
$ALOC1
$ASUMR
$CLKHD
$CRAVL
$DEACB
$DEAC1
$DEUMR
$DEVHD
$EXSIZ
$FMASK
$NS0
$NS1
$NS2
$NS3
$NS4
$NS5
$NS6
$NS7
$PARHD
$TKTCB
.TT0

NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES
U.SCB	000020	G	5-108
U.STS	000005	G	5-108
U.UNIT	000006	G	5-108
$ALOCB	= ******	G	5-114
$ALOC1	= ******	G	5-114
$ASUMR	= ******	G	5-117
$CLKHD	= ******	G	5-115
$CRAVL	= ******	G	5-114
$DEACB	= ******	G	5-114
$DEAC1	= ******	G	5-114
$DEUMR	= ******	G	5-117
$DEVHD	= ******	G	5-114
$EXSIZ	= ******	G	5-114
$FMASK	= ******	G	5-114
$NS0	= ******	G	5-115
$NS1	= ******	G	5-115
$NS2	= ******	G	5-115
$NS3	= ******	G	5-115
$NS4	= ******	G	5-115
$NS5	= ******	G	5-115
$NS6	= ******	G	5-115
$NS7	= ******	G	5-115
$PARHD	= ******	G	5-114
$TKTCB	= ******	G	5-115
.TT0	= ******	G	5-115

NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 3

POLAL -
$CEACX

341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396

```
                                        .SBTTL  $CEACX - ACCESS BLOCK IN EXTENDED POOL

                                ;+
                                ; **-$CEACX-ACCESS BLOCK IN EXTENDED POOL
                                ;
                                ; THIS SUBROUTINE MAPS TO A BLOCK IN EXTENDED SINGLE WORD POOL,
                                ; AND RETURNS A VIRTUAL ADDRESS TO ACCESS THE BLOCK.
                                ;
                                ; INPUTS:
                                ;       2(SP) = ADDRESS OF BLOCK TO BE ACCESSED
                                ;
                                ; OUTPUTS:
                                ;       2(SP) = VIRTUAL ADDRESS OF MAPPED BLOCK
                                ;
                                ; NOTE: MAPPING WILL BE MODIFIED IF THE BLOCK IS WITHIN THE EXTENDED POOL
                                ;       (KISAR5 FOR M/S/M-PLUS; UPAR0+4 FOR IAS)
                                ;-


000470                          $CEACX::
                                        .IF DF  M$$MGE
                                .
000470  032766  000001  000002          BIT     #1,2(SP)        ; IS THE BLOCK IN EXTENDED POOL ?
000476  001427                          BEQ     100$            ; IF EQ, NO - ALREADY MAPPED TO IT
000500                                  SAVRG   <R0>            ; ELSE, SAVE R0
000502  016600  000004                  MOV     4(SP),R0        ; COPY BLOCK ADDRESS
000506  042766  177701  000004          BIC     #177701,4(SP)   ; RETURN VIRTUAL ADDRESS

                                        .IF DF  R$$11M

000514  052766  120000  000004          BIS     #120000,4(SP)   ; ... MAPPED THROUGH KISAR5

                                        .IFF

                                        BIS     #040000,4(SP)   ; ... MAPPED THROUGH APR2

                                        .ENDC

                                        .IF DF  R$$EIS

                                        ASH     #-6,R0

                                        .IFF

                                        .REPT   6
                                ROR     R0              ; CONVERT ADDRESS TO BLOCK BIAS
                                        .ENDR

                                        .ENDC

000536  042700  176000                  BIC     #176000,R0      ; CLEAR EXTRANEOUS BITS


000542  017777  0000006 0000006         MOV     @XBIAS,@KSAR5   ; MAP TO BASE ADDRESS OF POOL
000550  060077  0000006                 ADD     R0,@KSAR5       ; ... UPDATE TO PROPER BLOCK

000554                                  RESRG   <R0>            ; RESTORE R0
```



```
PRALL

SYMBOL

SYMBOL

HEADR
I$$AS
M$$MGE
R$$11D
R$$11M
Z.AVL
$ALL16
$ALOC2
$ALPRO
$ALP16
$BIAS
$MAPX
$PDVA
$PRAVL
```

PRALL CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
HEADR	= ******	GX	6-104		
I$$AS	= ******		5-60		
M$$MGE	= 000000		6-86		
R$$11D	= ******		5-60		
R$$11M	= 000000		6-80		
Z.AVL	= ******	GX	6-84	*6-113	*6-114
$ALL16	= ******	GX	7-166		
$ALOC2	= ******	GX	6-93		
$ALPRO	000106	RG	#7-163		
$ALP16	000000	RG	#6-79	7-164	
$BIAS	= ******	GX	6-89		
$MAPX	= ******	GX	6-90		
$PDVA	= ******	GX	6-83		
$PRAVL	= ******	GX	*6-84	6-92	6-113

PRIO
PR.LB
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
4
4
4

```
                                ;
                                ;
000454  103534                12$:    BCS     142$            ; IF ECL MISSING - BRANCH
000456  005767  000004'              TST     LNKS            ; ENDNODE ?
000462  100443                       BMI     10$             ; IF NO - BRANCH
                                ;
000464                                SAVRG   <R1>            ; SAVE REG
000466  016700  000004'               MOV     LNKS,R0         ; GET MAX LINKS
000472  012701  000012                MOV     #C$LEN,R1       ; CALCULATE NI CACHE ENTRY LENGTH
000476                                CALL    $MUL            ; ...
000502  010100                        MOV     R1,R0           ; ...
000504                                RESRG   <R1>            ; RESTORE REG
                                ;
000506  062700  000077                ADD     #77,R0          ; ROUND UP TO NEXT BLOCK
000512  042700  000077                BIC     #77,R0          ; ...
                                ;
                                ;       .IF DF M$$MGE
                                ;
000516                                ASR$    6,R0            ; CONVERT BYTES TO BLOCKS
                                ;
                                ;       .IFTF   ; DF M$$MGE
                                ;
000532  060002                        ADD     R0,R2           ; INCREASE EXTEND SIZE
000534  060067  000020'               ADD     R0,$LDSIZ       ; AND LOAD SIZE
                                ;
000540                          9$:   .IFT    ; DF M$$MGE
                                ;
000540  016700  000006'               MOV     EXTEND,R0       ; GET THE EXTEND SIZE
000544  062700  000077                ADD     #77,R0          ; ROUND UP TO NEXT BLOCK
000550  042700  000077                BIC     #77,R0          ; ...
                                ;
000554                                ASR$    6,R0            ; CONVERT BYTES TO BLOCKS
000570  060002                        ADD     R0,R2           ; INCREASE THE EXTEND SIZE
                                ;
                                ;       .ENDC   ; DF M$$MGE
                                ;
000572  010167  000000'         10$:  MOV     R1,BLOX         ; SET PROCESS SIZE IN BLOCKS
000576  060267  000000'               ADD     R2,BLOX         ;
000602                                ASL$    6,R1            ; CONVERT FROM BLOCKS TO BYTES
000616                                ASL$    6,R2            ; BY SHIFTING LEFT 6 PLACES
000632  010167  000000G               MOV     R1,$LLEN        ; SET TRANSFER SIZE IN BYTES
000636  060102                        ADD     R1,R2           ; ADD LOAD SIZE TO EXTEND SIZE
000640  010267  000002'               MOV     R2,BYTES        ; SET PROCESS SIZE IN BYTES
                                ;
                                ;       .IF DF  M$$MGE
                                ;
000644  022767  114224  000000G       CMP     #^RXPT,$NAME    ; XPT ?
000652  001403                        BEQ     20$             ; IF YES - BRANCH
                                ;
000654  022702  020000                CMP     #20000,R2       ; IS PROCESS EXTENSION > 4K WORDS?
000660  103427                        BLO     141$            ; IF LO, YES
000662                          20$:
                                ;       .ENDC
                                ;
000662  012700  000000'               MOV     #DSKBUF,R0      ; RESTORE ADDRESS OF LABEL BLOCK BUFFER
000666  000241                        CLC                     ; INDICATE SUCCESS
000670                                RETURN
```

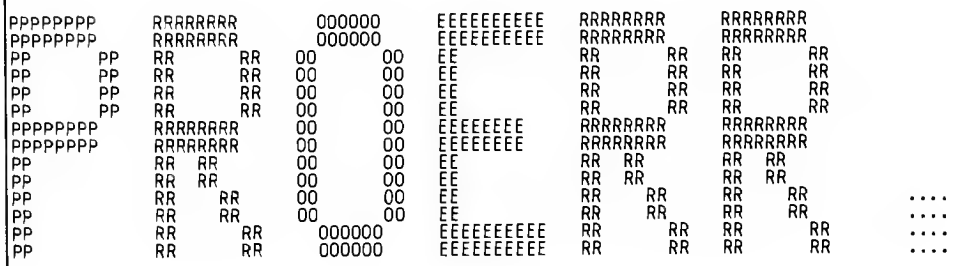
PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

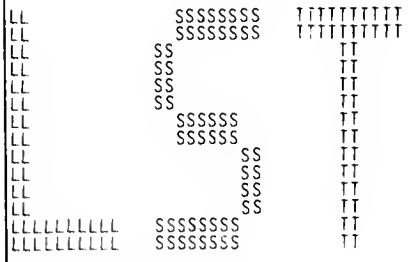
SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
M$$MGE	= 000000	5-74 7-153 9-301 9-394 9-422 9-443 10-495 10-571 11-596
PARHD	= ****** GX	10-481
PCBA	000012 R	#7-166 *8-221 8-238 *8-242 8-246 *10-551 13-760
PIRQ	= 177772	#5-74
PLGTH	= ****** GX	11-598 11-602 11-666
PMODE	= 030000	#5-74
PR.LBL	000156 R	8-228 #9-285
PR.PAR	000752 R	8-231 #10-475
PR.SUB	001332 R	10-501 #11-598
PR.XFR	001620 R	8-233 #13-758
PR0	= 000000	#5-74
PR1	= 000040	#5-74
PR2	= 000100	#5-74
PR3	= 000140	#5-74
PR4	= 000200	#5-74
PR5	= 000240	#5-74
PR6	= 000300	#5-74
PR7	= 000340	#5-74
PS	= 177776	#5-74
PS.COM	= 000200	10-539 11-619
PS.NSF	= 000400	11-619
PS.SYS	= 000040	10-497 11-619
P.BLKS	000016	10-510 10-527 *11-609 11-640 11-643
P.BUSY	000024	10-541 10-544 10-546 *10-546 10-547 *10-547
P.MAIN	000012	10-543 *11-608
P.NAM	000004	10-484 10-486 *11-606 *11-607
P.PRO	= ****** GX	*8-251
P.REL	000014	10-505 11-626 11-629 *11-634 *11-638 11-642 *11-651 13-761
P.STAT	000030	10-497 10-539 *11-619
P.SUB	000010	11-627 *11-635 *11-652 *11-655
R$$EIS	= ******	9-396 9-409 9-416 9-417 10-507
R$$MPL	= ******	5-74 5-74 5-74 10-489 11-611
R$$11D	= ******	5-74 13-773
R$$11M	= 000000	7-162 10-474 13-759
SWR	= 177570	#5-74
TPS	= 177564	#5-74
TS$NHD	= 040000	9-290
UBMPR	= 170200	#5-74
UISAR0	= 177640	#5-74
UISAR1	= 177642	#5-74
X$$MDB	= ******	10-513
$CLDEQ	= ****** GX	8-234
$CLIOS	= ****** GX	8-224 9-439 13-792
$CLOPE	= ****** GX	8-223
$FLAGS	= ****** GX	9-312 9-333 10-499 10-503 11-636 11-647
$IOSB	= ****** GX	9-439 13-792
$LDSIZ	000020 RG	#7-184 *9-295 *9-401
$LLEN	= ****** GX	*9-418
$MUL	= ****** GX	9-330 9-343 9-387
$NAME	= ****** GX	9-304 9-363 9-424 11-607
$PNAME	000014 RG	#7-171 *10-475 *10-476 *10-478 *10-479 10-484 10-486
$PRCLN	= ****** GX	9-342

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 3

FILEID**PROERR





PROERR CREATED BY MACRO ON 29-JUN-85 AT 00:37 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	7-168	7-172	7-185	7-188	8-205	8-213	9-236	9-243	9-246	
NTLER$	#5-59	6-75	6-76	6-77	6-78	6-79	6-80	6-81	6-82	6-83
	6-84	6-85	6-87	6-88	6-89	6-91	6-92	6-93	6-94	6-95
	6-97	6-98	6-99	6-101	6-102	6-103	6-104	6-105	6-106	6-107
	6-108	6-119	6-120	6-121	6-122					
RETURN	7-190	8-221	9-251							

FILEID**PROXPT

G 13

```
PROXPT - XPT PROCESS INITIALIZE MACRO V05.03b Saturday 29-Jun-85 00:37  Page 9-2
Symbol table

Size of core pool: 15496 words  ( 59 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:15.55
SY:PROXPT.V2,[132,134]PROXPT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]PROXPT
```

H 13

```
PROXPT     CREATED BY  MACRO  ON 29-JUN-85 AT 00:38     PAGE 1
```

```
SYMBL
SYMBOL
SYMBOL
CHECK
CL.OPE
E.BAD
E.OPEN
E.ORUN
E.REAL
F.DFNB
IE.EOV
IE.RBO
LOOK
N.FTYP
READ
SB.DEV
SB.LNK
SB.SYM
SB.VAL
STBBF
$CLBUF
$CLDEV
$CLFDB
$CLFLG
$CLIOS
$CLQUE
$CLSAV
$CLSIZ
$GCLEN
$SAVAL
$SYMBL
```

SYMBL CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES			
CHECK	000326 R	6-139	#8-219		
CL.OPN	= ****** GX	6-114			
E.BAD	000010	#5-68	7-200		
E.OPEN	000002	#5-65	6-164		
E.ORUN	000006	#5-67	7-194		
E.READ	000004	#5-66	7-197		
F.DFNB	= ****** GX	6-116			
IE.EOF	= ****** GX	7-198			
IE.RBG	= ****** GX	7-195			
LOOK	000372 R	6-146	#9-249		
N.FTYP	= ****** GX	6-117			
READ	000222 R	6-137	#7-181		
SB.DEF	000002	#5-77	*6-127	6-144	*9-265
SB.LNK	000000	#5-76			
SB.SYM	000004	#5-78	9-260	9-262	
SB.VAL	000010	#5-79	*6-128	*9-266	
STBBF	000000 R	#5-91	7-181	7-188	
$CLBUF	= ****** GX	*7-181			
$CLDEQ	= ****** GX	6-153			
$CLFDB	= ****** GX	6-116			
$CLFLG	= ****** GX	*6-114			
$CLIOS	= ****** GX	6-134	7-184		
$CLQUE	= ****** GX	6-133			
$CLSAV	= ****** GX	6-113			
$CLSIZ	= ****** GX	*7-182			
$GCLEP	= ****** GX	7-183			
$SAVAL	= ****** GX	6-115			
$SYMBL	000000 RG	#6-112			

SYMBL CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 2

```
                                        ;+
                                        ; $LFLHD - ".LFLHD" ACTION ROUTINE
                                        ; $MPLHD - ".MPLHD" ACTION ROUTINE
                                        ; $LSTHD - ".LSTHD" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       $LFLD/$MPLD=CURRENT VALUE OF $SIZE
                                        ;-
                                        
                                                .ENABL  LSB
                                        
000416                                  $LFLHD::
000416  016767  000000G 000000G                 MOV     $SIZE,$LFLD     ; SAVE CURRENT DISPLACEMENT
000424  012700  000000G                         MOV     #F1.LFL,R0      ; GET $FLAG1 BIT CODE
000430  000407                                 BR      10$
000432                                  $MPLHD::
000432                                          CALL    $VFDLC          ; MAKE SURE IT'S A DLC
000436  016767  000000G 000000G                 MOV     $SIZE,$MPLD     ; SAVE CURRENT DISPLACEMENT
000444  012700  000000G                         MOV     #F1.MPL,R0      ; GET $FLAG1 BIT CODE
000450                                  10$:    CALL    $VFSPC          ; MAKE SURE THIS IS THE FIRST AND ONLY
000454                                  $LSTHD::
000454                                          CALL    $RQ1            ; REQUEST BINARY FILE
000460  103405                                  BCS     20$             ; IF CS, NO ACCESS
000462  112712  000000G                         MOVB    #.LSTHD,(R2)    ; STORE FUNCTION CODE
000466  062767  000004  000000G                 ADD     #4,$SIZE        ; ADJUST DATA BASE SIZE
000474                                  20$:    RETURN
                                        
                                                .DSABL  LSB
```

```
                                        ;+
                                        ; $VXLLC - VERIFY NOT LLC ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       ERROR IF LLC BEING LOADED
                                        ;-
001434                                  $VXLLC:
001434  032767  000000C 000000G                 BIT     #FL.DDM!FL.DLC,$FLAGS ; IS IT EITHER DDM OR DLC?
001442  001003                                  BNE     10$             ; IF NE, YES
001444                                          EMSG$   80              ; DDM/DLC OPERATOR IN LLC TEMPLATE
001452                                  10$:    RETURN

000001                                          .END
```

```
                                        .SBTTL  DEAECL - DEALLOCATE ECL NODE COUNTERS

                                ;+
                                ;
                                ; DEAECL - DEALLOCATE ECL NODE COUNTER BLOCK
                                ;
                                ; INPUTS:
                                ;       R3 - ADDRESS OF LINE TABLE
                                ;       R4 - ADDRESS OF ECL'S PDV
                                ;
                                ; OUTPUTS:
                                ;       R3 - PRESERVED
                                ;       R4 - PRESERVED
                                ;       ECL NODE COUNTER BLOCK IS DEALLOCATED
                                ;
                                ;-
000422  016301  000044          DEAECL: MOV     N$ENC+2(R3),R1          ; GET START OF LAST NODE COUNTER BLOCK
000426  062701  000050                  ADD     #E$NLEN,R1              ; POINT TO END OF COUNTER BLOCK
000432  166301  000042                  SUB     N$ENC(R3),R1            ; CALCULATE NUMBER OF BYTES
000436  062701  000077                  ADD     #77,R1                  ; ROUND TO NEXT BLOCK BOUNDARY
000442  042701  000077                  BIC     #77,R1                  ; ...
000446                                  ASR$    6,R1                    ; CONVERT TO BLOCK COUNT
000462  016300  000046                  MOV     N$ENC+4(R3),R0          ; GET APR BIAS
000466  001402                          BEQ     10$                     ; NO COUNTERS ALLOCATED
000470                                  CALL    $DEA18                  ; DEALLOCATE MINHOP/COST MATRIX
000474                          10$:    RETURN
```

```
                                        .SBTTL  ICBLEN - CHECK ICB FOR LENGTH
                                ;+
                                ; *** - ICBLEN - CHECK ICB FOR LENGTH
                                ;
                                ; INPUT:
                                ;       R2 = START ADDRESS OF LINE TABLE
                                ;       R3 = END ADDRESS OF LINE TABLE
                                ;       R4 = START ADDRESS OF ASSUMED ICB
                                ;       $BIAS = BIAS FOR PROCESS (FOR NON RSX-11M+ SYSTEMS)
                                ;       BIAS = BIAS FOR PROCESS (RSX-11M+)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;               IF SUCCESS:
                                ;                       R3 = LENGTH OF ICB BLOCK
                                ;                       ICBSZ = SIZE OF ICB BLOCK
                                ;       R4,R5 = DESTROYED
                                ;
                                ;-
001720                          ICBLEN:
001720  005005                  10$:    CLR     R5              ;; CLEAR NUMBER OF MATCHES FOUND
001722  022724  013746                  CMP     #13746,(R4)+    ;; IS IT REALLY AN ICB?
001726  001013                          BNE     20$             ;; IF NE, NO
001730  022424                          CMP     (R4)+,(R4)+     ;; SKIP TO BIAS
                                        .IF DF  R$$MPL
                                        MOV     FMASK,FMKTMP    ;; GET $FMASK+2 ADDRESS
                                        ADD     #2,FMKTMP       ;; ...
                                        BIT     #F2.DAS,@FMKTMP ;; EXTRA APR SAVED ?
                                        BEQ     15$             ;; IF EQ, NO
                                        CMP     (R4)+,(R4)+     ;; SKIP TO REAL BIAS
                                15$:    CMP     BIAS,(R4)+      ;; BIAS MATCH?
                                        .IFF
001732  026724  000000G                 CMP     $BIAS,(R4)+     ;; BIAS MATCH?
                                        .IFTF
001736  001007                          BNE     20$             ;; IF NE, NO - NOT AN ICB
001740  022424                          CMP     (R4)+,(R4)+     ;; SKIP TO LINE TABLE ADDRESS
                                        .IFT
                                        BIT     #F2.DAS,@FMKTMP ;; IS DATA SPACE ENABLED ?
                                        BEQ     17$             ;; IF EQ, NO - NO EXTRA APR TO SET UP
                                        ADD     #6,R4           ;; SKIP TO LINE TABLE ADDRESS
                                17$:
                                        .ENDC
001742  021402                          CMP     (R4),R2         ;; JMP ADDRESS INSIDE LINE TABLE?
001744  103404                          BLO     20$             ;; IF LO, NO
001746  022403                          CMP     (R4)+,R3        ;; JMP ADDRESS INSIDE LINE TABLE?
001750  101002                          BHI     20$             ;; IF HI, NO
                                ;
```

```
                                        .SBTTL  XALITB - DE-ALLOCATE ALIAS TABLE

                                ;+
                                ;
                                ; XALITB - DE-ALLOCATE THE ALIAS TABLE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ALL ALIAS NAME BLOCKS ARE DE-ALLOCATED
                                ;
                                ;-

003142  017703  000000G       XALITB: MOV     @DECPT,R3       ; POINT AT DECNET HOME BLOCK
003146  001434                        BEQ     30$             ; IF PSI ONLY SYSTEM - BRANCH
003150  062703  000000                ADD     #D$ANN,R3       ; POINT AT ALIAS
003154                                SWSTK$  30$             ;; ENTER SYSTEM STATE
003160                                SAVMAP                  ;; SAVE CURRENT MAPPING
003164  011300                10$:    MOV     (R3),R0         ;; GET THE NEXT OBJECT IN LIST
003166  001422                        BEQ     20$             ;; IF EQ, END OF LIST
003170  010046                        MOV     R0,-(SP)        ;; SET BLOCK ADDRESS
003172                                CALL    $CEACX          ;; GET ACCESS TO BLOCK
003176  012604                        MOV     (SP)+,R4        ;; GET MAPPED ADDRESS OF BUFFER
003200  011413                        MOV     (R4),(R3)       ;; GET THE NEXT ALIAS IN LIST
003202  010401                        MOV     R4,R1           ;; COPY ADDRESS OF ALIAS BLOCK
003204  062701  000020                ADD     #A.ACC,R1       ;;      POINT AT START OF ACC. INFO.
003210  112102                        MOVB    (R1)+,R2        ;;      LENGTH OF FIRST FIELD
003212  060201                        ADD     R2,R1           ;;      ...
003214  112102                        MOVB    (R1)+,R2        ;;      LENGTH OF SECOND FIELD
003216  060201                        ADD     R2,R1           ;;      ...
003220  112102                        MOVB    (R1)+,R2        ;;      LENGTH OF THIRD FIELD
003222  060201                        ADD     R2,R1           ;;      ...
003224  160401                        SUB     R4,R1           ;;      TOTAL LENGTH OF ALIAS BLOCK
003226                                CALL    $XDEAC          ;; DE-ALLOCATE THE ALIAS BLOCK
003232  000754                        BR      10$             ;; AND GO AGAIN
003234                        20$:    RESMAP                  ;; RESTORE PREVIOUS MAPPING
003240                        30$:    RETURN
```

XSERTB DEALLOCATE SERVICE BLOCKS

```
2045 004606  062601                    ADD     (SP)+,R1        ; GET LAST ADDRESS OF BLOCK
2046
2047 004610  160401                    SUB     R4,R1           ; CALCULATE BLOCK LENGTH
2048 004612                            CALL    $XDEAC          ; DEALLOCATE FILENAME BLOCK
2049 004616  000752                    BR      10$             ; GET NEXT BLOCK
2050
2051 004620                    20$:    RESMAP                  ; RESTORE MAPPING
2052 004624                    30$:    RETURN
2053
2054
2055
2056         000001                    .END
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES					
S.NMST	000002	#5-72					
S.OWNR	000003	#5-72	*11-428	*18-1043			
S.PSS	000022	#5-74					
S.SEC	000040	#5-74					
S.TER	000042	#5-74					
T$FLAG	000044	#5-64					
T$LIF	000013	#5-64					
T$LIFL	000013	#5-64					
T$LIFO	000013	#5-64					
T$LIFS	000013	#5-64					
T$LIN	000000	#5-64					
T$LIPS	000006	#5-64					
T$LLD	000012	#5-64					
T$LLDC	000045	#5-64					
T$LLDL	000012	#5-64					
T$LLDO	000012	#5-64					
T$LLDS	000012	#5-64					
T$LLEN	000046	#5-64					
T$LOPR	000002	#5-64					
T$LTCL	000024	#5-64					
T$LTIM	000026	#5-64					
T$LTPR	000014	#5-64					
T$LTPS	000020	#5-64					
T$NAPL	000004	#5-64					
T$NFE	000000	#5-64					
T$NLEN	000010	#5-64					
T$NNUL	000002	#5-64					
T$NOPL	000006	#5-64					
T$NRNI	000042	#5-64					
T$NRPL	000005	#5-64					
T$NRUL	000007	#5-64					
T$NVR	000001	#5-64					
T$RPRI	000040	#5-64					
T$SVC	000034	#5-64					
T$T5	000030	#5-64					
T$T6	000032	#5-64					
T1SCL =	****** GX	19-1083					
T100C =	****** GX	13-539					
T100Q =	****** GX	13-517	13-531				
UBA	000012 R	#7-148	15-699	*27-1572	*27-1630	28-1665	
UBIAS	000010 R	#7-147	*27-1633	28-1716	28-1730	28-1745	28-1763
UPTR	000114 R	#7-158	*28-1670	28-1710	28-1755		
URMTMP	000004 R	#7-136					
VECT	001412 RG	14-621	#15-679				
XALITB	003142 R	19-1154	19-1297	#24-1473			
XDCB	003046 R	12-493	19-1123	#20-1364			
XFILTB	004522 R	19-1158	#32-2020				
XFLTTB	003272 R	19-1156	19-1299	#26-1537			
XF.CAN =	000200	#5-77					
XF.DDM =	000001	#5-77					
XF.DLC =	000002	#5-77					
XF.LLC =	000004	#5-77					

NTL11M CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES	
DCBDF$	#5-58	5-67
HWDDF$	#5-58	5-73
LBLDF$	#5-58	5-80
PCBDF$	#5-58	5-86
SCBDF$	#5-58	5-95
TCBDF$	#5-58	5-101
UCBDF$	#5-58	5-107

FILEID**POLAL

```
                                        .ENDC
                                
000556                          100$:   RETURN                  ; RETURN TO THE CALLER
                                
000001                                  .END
```

PRALL CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME REFERENCES

CALL	6-82	6-90	6-93	7-164	7-166
RETURN	6-125	7-167			
SWSTK$	6-82				

FILEID**PRIO

```
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000672  116777  000000G 000000G 101$:   MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
000700  012767  000010  000010'         MOV     #ERR8,ERROR     ; LABEL BLOCK READ FAILURE
000706  000417                          BR      142$
000710  012767  000012  000010' 111$:   MOV     #ERR10,ERROR    ; INVALID PROCESS IMAGE
                                        .IF DF  M$$MGE
000716  000413                          BR      142$
000720  012767  000014  000010' 121$:   MOV     #ERR12,ERROR    ; PROCESS NOT BUILT FOR APR5
000726  000407                          BR      142$
000730  012767  000016  000010' 131$:   MOV     #ERR14,ERROR    ; PROCESS EXCEEDS 4K WORDS
000736  000403                          BR      142$
000740  012767  000020  000010' 141$:   MOV     #ERR16,ERROR    ; PROCESS EXTENSION EXCEEDS 4K WORDS
                                        .ENDC
000746  000261                  142$:   SEC                     ; INDICATE FAILURE
000750                                  RETURN
```

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

```
SYMBOL  VALUE           REFERENCES

$PREXT  = ******  GX     9-338
$PRINC  = ******  GX     9-325
$PRIO     000000 RG     #8-220
$PRLLN  = ******  GX     9-329
$READX  = ******  GX     13-770
$RLBL   = ******  GX     9-286
$XPAR   = ******  GX     8-240
$$EXT   = ******  GX     9-341       9-352
$$INC   = ******  GX     9-328
$$PAR   = ******  GX     10-475      10-476
.BASEB  = 140000        #5-74
```

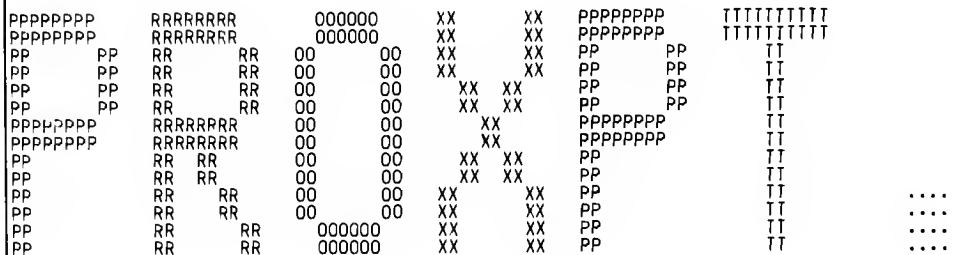

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 4

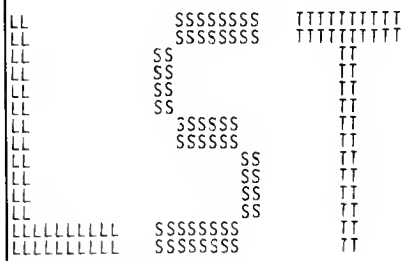
H 11
PROERR - PROCESS RELATED ERRORS MACRO V05.03b Saturday 29-Jun-85 00:37
Table of contents

5- 55 MACRO DEFINITIONS
6- 61 LOCAL DATA
7- 156 PERR - PRINT EPROR MESSAGE
8- 192 REP4A - REPLACE PROCESS NAME
9- 223 REP5A - REPLACE PARTITION NAME
10- 253 REP5B - REPLACE REGION NAME

I 11
PROERR - PROCESS RELATED ERRORS MACRO V05.03b Saturday 29-Jun-85 00:37 Page 4

```
**FILE**ID**PROXPT
```






```
PROXPT - XPT PROCESS INITIALIZE MACRO V05.03b Saturday 29-Jun-85 00:37
```

PROXPT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
A$LEN	000004		7-158								
C$LEN	000012		6-104								
DECPT	= ******	GX	6-93	6-100	7-149	8-226					
D$LNUM	000014		7-185								
D$MLL	000040		6-102								
D$NA	000062		8-230	8-234	8-255						
D$NBEA	000056		7-157								
D$NBRA	000054		7-156								
D$NN	000060		6-94	7-150	8-227	8-249					
E$NBR	000014		#5-67								
E$NBS	000020		#5-67								
E$NCR	000034		#5-67								
E$NCS	000036		#5-67								
E$NIC	000044		#5-67								
E$NLEN	000050		#5-67								
E$NLLA	000012		#5-67								
E$NLNK	000000		#5-67								
E$NML	000040		#5-67								
E$NMR	000024		#5-67								
E$NMS	000030		#5-67								
E$NNOD	000002		#5-67								
E$NRT	000042		#5-67								
E$NRTP	000005		#5-67								
E$NSEG	000010		#5-67								
E$NTIM	000046		#5-67								
E$NUSE	000004		#5-67								
E$STRT	000006		#5-67								
I$$AS	= ******		7-164	7-196	8-240	9-331					
KSAR5	= ******	GX	6-109	6-112	6-116	6-123	6-128	8-282	8-286	8-290	8-298
			8-302								
M$$MGE	= 000000		8-253	9-339							
N$ADJ1	000072		*7-172	*7-205							
N$ADJ2	000074		7-182	7-191	*7-203	*7-206					
N$CACH	000062		*6-107	6-110	*6-131	*6-132					
N$ROA1	000022		*8-248	*8-249	*8-250	*8-254	8-258	8-261	8-262	8-267	8-274
N$ROA2	000030		*8-258	*8-259	*8-260	*8-261	*8-264	8-275			
N$TLC	000100		*8-307	9-324	*9-337	*9-340	9-341	9-348			
N$TNC	000114		8-283	8-284							
R$$EIS	= ******		7-164	7-196	8-240	9-331					
R$$T1D	= ******		7-164	7-196	8-240	9-331					
R$$T1M	= 000000		8-253	9-339							
T$FLAG	000044		#5-67								
T$LIF	000013		#5-67								
T$LIFL	000013		#5-67								
T$LIFO	000013		#5-67								
T$LIFS	000013		#5-67								
T$LIN	000000		#5-67								
T$LIPS	000006		#5-67								
T$LLD	000012		#5-67								
T$LLDC	000045		#5-67								
T$LLDL	000012		#5-67								
T$LLDO	000012		#5-67								

PROXPT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 2

SYMBL CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES							
CALL	6-113	6-115	6-133	6-137	6-139	6-146	6-153	7-183
RETURN	6-157	7-189	7-202	8-233	9-268			

FILEID**TMPACT

```
                                ;+
                                ; $LTAB - ".LTAB" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000476                          $LTAB::
000476                                  CALL    $RQ1            ; REQUEST BINARY BUFFER ACCESS
000502  103405                         BCS     10$             ; IF CS, ACCESS DENIED
000504  112712  000000G                MOVB    #.LTAB,(R2)     ; STORE FUNCTION CODE
000510  062767  000002  000000G        ADD     #2,$SIZE        ; ADJUST LINE TABLE SIZE
000516                          10$:    RETURN
```

Symbol table

```
A$$CHK= 000000          LF.REA= 010000          N$$ACC= 000001          $DPRB   000742RG          $SZPTR= ****** GX
A$$CPS= 000000          LF.SER= 000040          N$$BUF= 000001          $DPRW   000770RG          $TERR = ****** GX
A$$PRI= 000000          LF.TIM= 000010          N$$LDV= 000001          $DVCHA  000230RG          $TIME   001002RG
A$$TRP= 000000          LF.UNL= 020000          N$$MCP= 000001          $ELINE= ****** GX         $TWARN= ****** GX
C$$CKP= 000000          LF.X2P= 000000          N$$MLL= 000001          $END1   000304RG          $UNB    000626RG
C$$ORE= 000400          LN.CLO= 000000          N$$MOV= 000010          $END2   000340RG          $UNW    000704RG
C$$RSH= 177564          LN.DUM= 000005          N$$NCT= 000001          $ERR14  000042R   002     $VALUE= ****** GX
D$$BUG= 177514          LN.LOA= 000004          N$$PEM= 000001          $ERR15  000106R   002     $VFDDM= ****** GX
D$$ISK= 000000          LN.LOO= 000003          P$$P45= 000000          $ERR18  000146R   002     $VFDLC= ****** GX
D$$L11= 000001          LN.DAU= 000003          P$$WRD= 000000          $ERR19  000212R   002     $VFLV1= ****** GX
D$$YNC= 000000          LN.OFF= 000001          Q$$OPT= 000010          $ERR2   000000R   002     $VFSPC= ****** GX
D$$YNM= 000000          LN.ON = 000000          R$$DER= 000000          $ERR20  000260R   002     $VXLLC  001434R
E$$XPR= 000000          LN.OOP= 000004          R$$K11= 000001          $ERR52  000314R   002     $WORD   001032RG
FL.DDM= ****** GX       LN.OPE= 000001          R$$SND= 000000          $ERR80  000370R   002     $ZBYTE  000076RG
FL.DLC= ****** GX       LN.REF= 000002          R$$T1M= 000000          $FILEN= ****** GX         $ZWORD  001026RG
FL.MUX= ****** GX       LN.SER= 000002          SF.ACT= 000200          $FLAGS= ****** GX         $$PRI = ****** GX
FMT2    000422R   002   LN.STA= 000017          SF.ENA= 000100          $FLAG1= ****** GX         .ADDRB= ****** GX
FM.2  = 000000          LN.SUB= 000360          SF.LPB= 000004          $FLAG2= ****** GX         .ADDRW= ****** GX
F$$LVL= 000001          LN.TRI= 000006          SF.MFL= 000040          $FNAME  001350RG          .APR  = ****** GX
F1.LFL= ****** GX       L$$ASG= 000000          SF.PAC= 000020          $LFLD = ****** GX         .BLKB = ****** GX
F1.MPL= ****** GX       L$$DRV= 000000          SF.REA= 000010          $LFLHD  000416RG          .BLKW = ****** GX
F2.BYT= ****** GX       L$$P11= 000001          SF.SER= 000001          $LSTHD  000454RG          .BYTE = ****** GX
F2.FIL= ****** GX       L$$11R= 000000          SF.SVC= 000002          $LTAB   000476RG          .CSR  = ****** GX
F2.NEG= ****** GX       L.COST  000015          SF.UNL= 000040          $MOD    001260RG          .CTIM = ****** GX
F2.SUB= ****** GX       L.CTL   000012          S$$WRG= 000000          $MPLD = ****** GX         .DPRB = ****** GX
G$$TPP= 000000          L.CVA   177776          S$$YSZ= 007600          $MPLHD  000432RG          .DPRW = ****** GX
G$$TSS= 000000          L.DDM   000002          S.COST  000001          $NOD    001064RG          .DVCHA= ****** GX
G$$TTK= 000000          L.DDS   000004          S.FLG   000000          $NOMDD  001342RG          .LSTHD= ****** GX
G$$WRD= 000000          L.DLC   000003          S.LEN   000004          $NUM    001154RG          .LTAB = ****** GX
I$$RAR= 000000          L.DLM   000006          S.NMST  000002          $ODD    000520RG          .NODE = ****** GX
I$$RDN= 000000          L.DLS   000010          S.OWNR  000003          $OFF  = ****** GX         .PECHA= ****** GX
K$$CNT= 177546          L.FLG   000000          T$$KMG= 000000          $PECHA  000256RG          .PNUMB= ****** GX
K$$CSR= 177546          L.KRBA  000016          T$$MIN= 000000          $PRI    000556RG          .PNUMH= ****** GX
K$$LDC= 000000          L.LEN = 000022          V$$CTR= 001000          $PRI1   000552RG          .PRI  = ****** GX
K$$TPS= 000074          L.MPF   000022          X$$DBT= 000000          $PROV   001106RG          .PRTOV= ****** GX
LD$LP = 000000          L.NMST  000020          $ADDRB  000604RG        $REP.C= ****** GX         .PSTCN= ****** GX
LF.ACT= 100000          L.NSTA  000014          $ADDRW  000654RG        $RQ1  = ****** GX         .PSTPT= ****** GX
LF.BRO= 000400          L.OWNR  000021          $APR    000734RG        $RQ2  = ****** GX         .SLNB = ****** GX
LF.BWT= 000007          L.UNT   000013          $BLKB   000000RG        $RQ3  = ****** GX         .SLNW = ****** GX
LF.ENA= 002000          MAXOV = ****** GX       $BLKW   000014RG        $SIZE = ****** GX         .STNB = ****** GX
LF.LPB= 001000          M$$CRB= 000124          $BYTE   000102RG        $SLNB   000620RG          .STNW = ****** GX
LF.MDC= 000100          M$$CRX= 000000          $CNB    000126RG        $SLNW   000676RG          .TIME = ****** GX
LF.MFL= 004000          M$$FCS= 000000          $CNW    000150RG        $SLTA = ****** GX         .UNB  = ****** GX
LF.MTP= 000020          M$$MGE= 000000          $CSR    000176RG        $STNB   000612RG          .UNW  = ****** GX
LF.PAC= 000200          M$$NET= 000000          $CTM    000662RG        $STNW   000670RG          .WORD = ****** GX
LF.RDY= 040000          M$$OVR= 000000          $DIFF = ****** GX

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001454    001   (RW,I,LCL,REL,CON)
DATA    000442    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
```


Symbol table

```
                                        .SBTTL  $XLIN - CLEAN UP LINE

                                ;+
                                ; $XLIN - CLEAN UP AFTER AN UNSUCCESSFUL "SET LINE"
                                ;
                                ; THIS ROUTINE IS CALLED IF A "SET LINE" FAILS IN TRANSFERRING EITHER THE
                                ; DLC OR DDM PROCESS INTO CORE, INITIALIZING THE DLC LINE TABLE, OR LOADING
                                ; THE DDM MICRO-CODE. IN NO CASE HAS THE DDM LINE TABLE BEEN INITIALIZED.
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;       $DLCPV=DLC PDV ADDRESS
                                ;       $DDMPV=DDM PDV ADDRESS
                                ;       FL.PX1 OF $FLAGS SET IF DLC WAS TRANSFERRED
                                ;       FL.PX2 OF $FLAGS SET IF DDM WAS TRANSFERRED
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;-
000476                          $XLIN::
000476  016705  000000G                 MOV     $SLTA,R5        ; GET SLT ADDRESS

                                        .IF DF  R$$MPL
                                        MOV     $DDMPV,R4       ; GET THE DDM PDV ADDRESS
                                        BEQ     111$            ; IF EQ, LOGIC ERROR
                                        CALL    FNCTB           ; FIND THE CTB ADDRESS
                                        MOVB    L.CTL(R5),R0    ; GET THE CONTROLLER NUMBER
                                        ASL     R0              ; CONVERT TO A WORD INDEX
                                        ADD     R0,R1           ; POINT AT KRB ADDRESS IN CTB
                                        CLR     L.KRB(R1)       ; ZERO KRB POINTER
                                        CALL    $HRCIN          ; RE-INITIALIZE HRC...
                                        .ENDC

000502  105065  000021                  CLRB    L.OWNR(R5)      ; CLEAR THE LINE OWNER
000506  116500  000014                  MOVB    L.NSTA(R5),R0   ; GET THE NUMBER OF STATIONS
000512  001411                          BEQ     3$              ; IF EQ, NONE
000514  010501                          MOV     R5,R1           ; POINT TO START OF TRIBUTARIES
000516  062701  000022                  ADD     #L.MPF,R1       ; ...
000522  105061  000003          1$:     CLRB    S.OWNR(R1)      ; CLEAR STATION OWNERS
000526  062701  000004                  ADD     #S.LEN,R1       ; NEXT STATION
000532                                  SOB     R0,1$           ; CLEAR ALL STATIONS

000536  016500  000016          3$:     MOV     L.KRBA(R5),R0   ; GET THE KRB ADDRESS
000542  001404                          BEQ     5$              ; IF EQ, THERE IS NONE
000544                                  CALL    XKRB            ; DE-ALLOCATE THE KRB
000550  005065  000016                  CLR     L.KRBA(R5)      ; CLEAR THE KRB POINTER

000554  016503  000010          5$:     MOV     L.DLS(R5),R3    ; AND DLC LINE TABLE ADDRESS
000560  001417                          BEQ     20$             ; IF ZERO, NO WORK TO DO
000562  032767  000000G 000000G         BIT     #FL.PX1,$FLAGS  ; WAS DLC TRANSFERRED INTO CORE ?
000570  001003                          BNE     10$             ; YES .. DISCARD PARTITION LINE TABLES
000572  016767  000000G 000152'         MOV     $DLCPV,..PDV    ; NO .. HOUSEKEEP THE DE-ALLOCATIONS

000600  116504  000003          10$:    MOVB    L.DLC(R5),R4    ; GET THE DLC PDV INDEX
000604                                  CALL    XLT16           ; DE-ALLOCATE DLC LINE TABLE
000610  005065  000010                  CLR     L.DLS(R5)       ; ZERO LINE TABLE ADDRESS
000614  005065  000006                  CLR     L.DLM(R5)       ; AND LINE TABLE BIAS
```

```
                                        ; VALID ICB FOUND
                                        ;
001752  005205                          INC     R5              ;; INCREMENT THE NUMBER FOUND
001754  000762                          BR      10$             ;; CONTINUE UNTIL NO MORE

001756  005705                  20$:    TST     R5              ;; ANY ICB'S FOUND?
001760  001411                          BEQ     50$             ;; IF EQ, NO
001762  005003                          CLR     R3              ;; CLEAR SIZE OF DE-ALLOCATION
001764  066703  000000G         30$:    ADD     $ICBLN,R3       ;; ADD IN SIZE OF EACH ICB
001770  005305                          DEC     R5              ;;  FOR EACH ICB FOUND
001772  003374                          BGT     30$             ;;
001774  010367  000006'                MOV     R3,ICBSZ        ;; SAVE THE ICB BLOCK SIZE
                                        .IF DF  R$$MPL
                                        ADD     #4,ICBSZ        ;; INCLUDE THE LINK POINTER AND COUNT FIELDS
                                        .ENDC
002000  000241                  40$:    CLC                     ;; CLEAR CARRY TO SHOW SUCCESS
002002  000401                          BR      60$             ;; AND LEAVE

002004  000261                  50$:    SEC                     ;; SHOW FAILURE - NOT AN ICB
002006                          60$:    RETURN
```

```
                                        .SBTTL  XREMTB - DE-ALLOCATE REMOTE TABLE

                                ;+
                                ;
                                ; XREMTB - DE-ALLOCATE THE REMOTE TABLE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ALL REMOTE NAME BLOCKS ARE DE-ALLOCATED
                                ;
                                ;-

003242  017703  000000G         XREMTB: MOV     @DECPT,R3       ; POINT AT DECNET HOME BLOCK
003246  001410                          BEQ     10$             ; IF PSI ONLY SYSTEM - BRANCH
003250  062703  000002                  ADD     #D$RNN,R3       ; CREATE REMOTE LISTHEAD ADDRESS
003254  011303                          MOV     (R3),R3         ;   ...
003256  001404                          BEQ     10$             ; IF EQ, COMM EXEC - NOT LOADED YET

003260  012701  000014                  MOV     #R.LEN,R1       ; GET LENGTH OF A REMOTE BLOCK
003264                                  CALL    DEALST          ; DEALLOCATE THE REMOTE BLOCK LIST
003270                          10$:    RETURN
```

Symbol table

```
AC$DNT= 000002          CF.EOM= 000004          CS.ROV= 000004          DL$SYN= 000005          EV$HCE= 035114
AC$X25= 000001          CF.HDR= 000020          CS.RSN= 010000          DQCLK   000766R       EV$HCI= 035113
AE$CIR= 000003          CF.LB = 100000          CS.SHU= 000001          D$AMXC  000072          EV$HFE= 000506
AE$LIN= 000001          CF.LIN= 000002          CS.SID= 000002          D$AMXH  000074          EV$IFL= 000413
AE$MOD= 000004          CF.SOM= 000010          CS.STR= 000004          D$ANN   000000          EV$IFO= 000415
AT$ACL= 000100          CF.SYN= 000040          CS.SUC= 000001          D$BRPR  000102          EV$IFS= 000414
AT$ADN= 000040          CF.TRN= 000100          CS.TMO= 020000          D$BRTM  000100          EV$INF= 000515
AT$AUP= 000020          CLKHD = ****** GX       CS.XUR= 000004          D$DELF  000045          EV$LDL= 000407
AT$CYC= 000004          CL$ASZ= 010500          CXOPT = ****** GX       D$DELW  000046          EV$LDN= 010416
AT$LV1= 000002          CL$DLL= 000500          C$$CKP= 000000          D$END = 000104          EV$LDO= 000411
AT$LV2= 000001          CL$ECL= 000300          C$$ORE= 000400          D$FNB   000034          EV$LDS= 000410
AT$NEX= 000200          CL$LDN= 010400          C$$RSH= 177564          D$HJOR  000024          EV$LSC= 000500
AT$UP = 000010          CL$MAN= 000000          C.ADD   000034          D$HOST  000022          EV$LUP= 000412
A$CIR   000003          CL$PAZ= 034100          C.BID   000003          D$INAC  000044          EV$NOL= 000402
A$LEN   000004          CL$PLH= 034000          C.BUF   000014          D$INCT  000042          EV$NRC= 000416
A$NID   000000          CL$PLL= 000600          C.BUF1  000014          D$IPL   000051          EV$NSC= 000200
A$TM    000003          CL$PRT= 034200          C.BUF2  000024          D$LID   000020          EV$NUL= 000401
A$TMI   000002          CL$ROU= 010000          C.CNT   000020          D$LNAM  000006          EV$NVR= 000406
A$TSZ   000000          CL$SES= 000200          C.CNT1  000020          D$LNUM  000014          EV$OPL= 000403
A$TYP   000002          CL$SGE= 035000          C.CNT2  000030          D$LST   000047          EV$PCC= 034000
A$$CHK= 000000          CL$SSE= 035100          C.FLG   000022          D$MAXC  000064          EV$PCI= 034001
A$$CPS= 000000          CL$TRN= 000400          C.FLG1  000022          D$MAXH  000066          EV$PCM= 034002
A$$PRI= 000000          CL$XL2= 013700          C.FLG2  000032          D$MAXV  000070          EV$PFE= 000404
A$$TRP= 000000          CL$XL3= 013600          C.FNC   000010          D$MLL   000040          EV$PPC= 034003
A.ACC   000020          CL$X2S= 013500          C.LIN   000006          D$MNOD  000041          EV$RCF= 000517
A.LNK   000000          CM.CIR= 000002          C.LNK   000000          D$NA    000062          EV$RDC= 010001
A.NAM   000002          CM.FMT= 100000          C.MOD   000011          D$NBEA  000056          EV$RDR= 010002
A.PUD   000010          CM.HRD= 000002          C.NSP   000004          D$NBRA  000054          EV$RJE= 035106
A.REM   000012          CM.LIN= 000000          C.PRO   000042          D$NEND= 000054          EV$RSC= 000501
A.UCB   000010          CM.LOO= 000001          C.RSV   000002          D$NLN   000030          EV$RUL= 000405
BLKSIZ= 000001          CM.XLO= 000004          C.STA   000007          D$NN    000060          EV$SNA= 035000
BLOCK   000014R     002 COUNT   000120R     002 C.STS   000012          D$OUTT  000043          EV$SNF= 000516
CB.CCB= 000002          CP.DCF= 000040          C.URM   177776          D$RETF  000050          EV$SPE= 035001
CB.DDM= 000040          CP.HDL= 000007          C.XACP  000004          D$RNN   000002          EV$XCE= 034110
CB.DLC= 000020          CP.PS = 177400          C.XID   000035          D$RTMR  000076          EV$XDI= 013600
CB.RDB= 000004          CP.PSI= 000200          C.XLEN  000044          D$SEG   000036          EV$XGW= 034111
CB.SDB= 000010          CP.XCF= 000100          C.XPLI  000040          D$SER   000032          EV$XMX= 000514
CB.SLI= 000100          CP.2FR= 000030          C.XPT   000034          D$SQRL  000052          EV$XRS= 000512
CB.XLB= 000001          CS.ABO= 000100          C.XSVC  000042          D$$BUG= 177514          EV$XSC= 000513
CCBLH = ****** GX       CS.BRO= 000002          C.XTC   000037          D$$ISK= 000000          EV$X2S= 013500
CCBSZ = ****** GX       CS.BUF= 000200          C.X25   000036          D$$L11= 000001          EV.CCB= 000001
CC.LLC= 000200          CS.CES= 000002          DDMAC   000124RG    002 D$$YNC= 000000          EV.CIR= 000020
CE.ABO= 100362          CS.CHN= 000010          DEAC2   004302R         D$$YNM= 000000          EV.LCB= 000100
CE.DAO= 100346          CS.CMP= 000200          DEAECL  000422R         D.DSP = ****** GX       EV.LIN= 000004
CE.DIS= 100366          CS.DCR= 000400          DEALST  004424RG        D.PCB = ****** GX       EV.MAP= 000002
CE.ERR= 100370          CS.DEF= 000004          DECPT = ****** GX       EF$ACT= 000001          EV.MOD= 000040
CE.ILN= 100350          CS.DEV= 000002          DEUMR = ****** GX       EV$ACF= 000201          EV.NOD= 000010
CE.LTO= 100356          CS.DIS= 000040          DEVHD = ****** GX       EV$ADR= 000420          EV.PRT= 000200
CE.MOP= 100372          CS.ENA= 000001          DLCAC   000122RG    002 EV$ADU= 000417          EXRQN = ****** GX
CE.NTE= 100361          CS.ENB= 000020          DL$AST= 000002          EV$APL= 000400          E$DATA  000020
CE.RTE= 100376          CS.ERR= 100000          DL$HLT= 000000          EV$ARC= 000421          E$EVTS  000000
CE.SRC= 100364          CS.FTL= 001000          DL$IST= 000001          EV$AUC= 000010          E$LCN   000016
CE.STP= 100352          CS.HCR= 000001          DL$MAI= 000004          EV$AUS= 000003          E$LIN   000000
CE.TME= 100354          CS.HFE= 002000          DL$OFF= 000001          EV$CDF= 000520          E$MOD   000012
CE.TMO= 100374          CS.LST= 040000          DL$ON = 000000          EV$COZ= 000011          E$NBR   000014
CE.UNS= 100344          CS.MTL= 004000          DL$RUN= 000003          EV$DBR= 000302          E$NBS   000020
CF.CHN= 000001          CS.RNG= 000010          DL$SHU= 000002          EV$GAS= 035101          E$NCR   000034
```



```
NTLUNL

SYMBOL

SYMBOL

XKRB
XLIN
XLT
XLT16
XOBJTB
XPRO
XREMTE
XSERTB
X.FLAG
X.ID
X.LINK
X.RTMR
X.TMR
ZDDS
ZF.INI
ZF.LLC
ZF.MFL
ZF.PSE
Z.AVL
Z.DAT
Z.DSP
Z.FLG
Z.NAM
Z.PCB
$BIAS
$CCEX
$CEAC
$CLIN
$CPRO
$DDMPV
$DEAHC
$DEA1C
$DEA1E
$DECEX
$DLCPV
$DSIN
$FLAG
$ICBLK
$MAPX
$MXVC
$NAME
$PCB
$PDVA
$PRAV
$SAVR
$SLTA
$XDEA
$XLBR
$XLIN
$XPAR
$XPRO
$XPSI
```

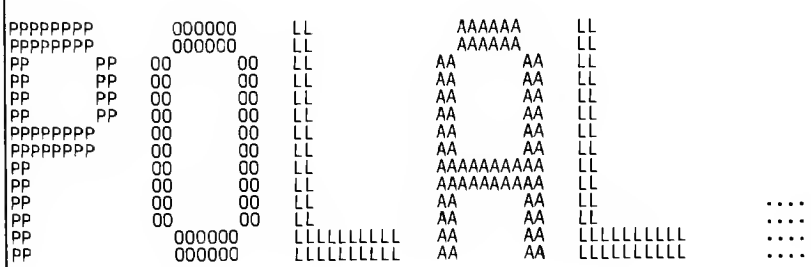
NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 10

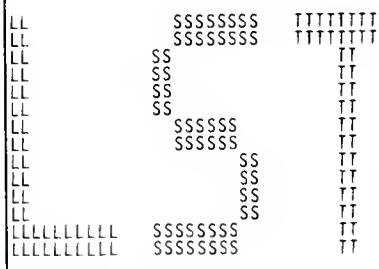
SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
XKRB	= ****** GX	11-434	14-626						
XLIN	001106 R	#14-568	19-1099						
XLT	003564 R	14-628	#28-1665						
XLT16	003560 R	8-307	11-444	14-647	#28-1664				
XOBJTB	003124 R	19-1153	19-1296	#23-1454					
XPRO	000022 R	#8-233	11-452	11-457	12-494	14-636	14-655	19-1124	
XREMTB	003242 R	19-1155	19-1298	#25-1512					
XSERTB	004476 R	19-1157	#32-1999						
X.FLAG	000003	#5-77							
X.ID	000002	#5-77	13-525						
X.LINK	000000	#5-77							
X.RTMR	000006	#5-77							
X.TMR	000004	#5-77							
ZDDS	002010 R	14-629	#18-1026						
ZF.INI	= ****** GX	8-233	12-483	19-1113	20-1364				
ZF.LLC	= ****** GX	8-266							
ZF.MFL	= ****** GX	12-495	14-634	14-653					
ZF.PSE	= ****** GX	14-572							
Z.AVL	= ****** GX	29-1851							
Z.DAT	= ****** GX	8-268	*8-309	9-340					
Z.DSP	= ****** GX	*8-264	*8-265	14-608					
Z.FLG	= ****** GX	8-233	8-266	12-483	*12-495	14-572	14-634	14-653	19-1113	20-1364
Z.NAM	= ****** GX	8-274	8-297	8-302	9-339	14-631				
Z.PCB	= ****** GX	8-238	*8-240	12-485	*12-491	19-1110	19-1117	*19-1121	20-1366	29-1824
$BIAS	= ****** GX	*14-608	17-972							
$CCEX	002134 RG	#19-1069								
$CEACX	= ****** GX	24-1481	26-1544	31-1975	32-2031					
$CLIN	001102 RG	#14-566								
$CPRO	000674 RG	#12-482								
$DDMPV	= ****** GX	11-455								
$DEAHC	= ****** GX	8-299								
$DEA16	= ****** GX	19-1139	19-1172	19-1199	19-1290	29-1891				
$DEA18	= ****** GX	10-387	28-1698	28-1718						
$DECEX	= ****** GX	15-870	19-1256	19-1333						
$DLCPV	= ****** GX	11-441	11-450							
$DSINI	000336 RG	#9-338	12-490	19-1120						
$FLAGS	= ****** GX	8-228	11-439	11-448	11-453					
$ICBLN	= ****** GX	17-1000								
$MAPX	= ****** GX	19-1195	19-1203	19-1272	27-1581	28-1725	28-1767	29-1847		
$MXVCT	= ****** GX	15-697								
$NAME	= ****** GX	*14-631								
$PCB	= ****** GX	8-230	8-235	*12-485	*19-1117	*20-1366	20-1372			
$PDVA	= ****** GX	8-227	12-482							
$PRAVL	= ****** GX	29-1850	29-1856							
$SAVRG	= ****** GX	18-1026								
$SLTA	= ****** GX	11-410	14-567	*19-1097						
$XDEAC	= ****** GX	24-1493	26-1549	31-1980	32-2048					
$XLBR	= ****** GX	28-1707								
$XLIN	000476 RG	#11-409								
$XPAR	= ****** GX	8-241	19-1223	19-1237	19-1308					
$XPRO	000000 RG	#8-226								
$XPSI	= ****** GX	19-1161	19-1163	19-1301						

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 11

```
**FILE**ID**POLAL
```


PPPPPPPP 000000 LL AAAAAA LL
PPPPPPPP 000000 LL AAAAAA LL
PP PP 00 00 LL AA AA LL
PP PP 00 00 LL AA AA LL
PP PP 00 00 LL AA AA LL
PP PP 00 00 LL AA AA LL
PPPPPPPP 00 00 LL AA AA LL
PPPPPPPP 00 00 LL AA AA LL
PP 00 00 LL AAAAAAAAAA LL
PP 00 00 LL AAAAAAAAAA LL
PP 00 00 LL AA AA LL ....
PP 00 00 LL AA AA LL ....
PP 000000 LLLLLLLLLL AA AA LLLLLLLLLL ....
PP 000000 LLLLLLLLLL AA AA LLLLLLLLLL ....




```
POLAL - NETWORK POOL ALLOCATION MACRO V05.03b Saturday 29-Jun-85 00:36
```

I 7

```
POLAL - NETWORK POOL ALLOCATION MACRO V05.03b Saturday 29-Jun-85 00:36  Page 10-2
Symbol table

ALOCB = ****** GX       E$$XPR= 000000          LD$LP = 000000          N$$LDV= 000001          S$$WRG= 000000
A$$CHK= 000000          F$$LVL= 000001          L$$ASG= 000000          N$$MCP= 000001          S$$YSZ= 007600
A$$CPS= 000000          G$$TPP= 000000          L$$DRV= 000000          N$$MLL= 000001          T$$KMG= 000000
A$$PRI= 000000          G$$TSS= 000000          L$$P11= 000001          N$$MOV= 000010          T$$MIN= 000000
A$$TFP= 000000          G$$TTK= 000000          L$$11R= 000000          N$$NCT= 000001          V$$CTR= 001000
C$$CkP= 000000          G$$WRD= 000000          M$$CRB= 000124          N$$PEM= 000001          XAVL  = ****** GX
C$$ORE= 000400          I$$RAR= 000000          M$$CRX= 000000          P$$P45= 000000          XAVLL   000002R
C$$RSH= 177564          I$$RDN= 000000          M$$FCS= 000000          P$$WRD= 000000          XBIAS = ****** GX
DEACB = ****** GX       KSAR5 = ****** GX       M$$MGE= 000000          Q$$DPT= 000010          X$$DBT= 000000
D$$BUG= 177514          K$$CNT= 177546          M$$NET= 000000          R$$DER= 000000          $CEACX  000470RG
D$$ISK= 000000          K$$CSR= 177546          M$$OVR= 000000          R$$K11= 000001          $LINKX  000424RG
D$$L11= 000001          K$$LDC= 000000          N$$ACC= 000001          R$$SND= 000000          $XALOC  000006RG
D$$YNC= 000000          K$$TPS= 000074          N$$BUF= 000001          R$$11M= 000000          $XDEAC  000134RG
D$$YNM= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000560    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 9106 Words  ( 36 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:11.07
SY:POLAL.V2,[132,134]POLAL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]POLAL
```

J 7

```
POLAL       CREATED BY  MACRO  ON 29-JUN-85 AT 00:36     PAGE 1
```

```
**FILE**ID**PRIO
```

```
PPPPPPPP      RRRRRRRR        IIIIII        000000
PPPPPPPP      RRRRRRRR        IIIIII        000000
PP       PP   RR       RR       II        00      00
PP       PP   RR       RR       II        00      00
PP       PP   RR       RR       II        00      00
PP       PP   RR       RR       II        00      00
PPPPPPPP      RRRRRRRR          II        00      00
PPPPPPPP      RRRRRRRR          II        00      00
PP            RR  RR            II        00      00
PP            RR  RR            II        00      00
PP            RR    RR          II        00      00        ....
PP            RR    RR          II        00      00        ....
PP            RR       RR    IIIIII      000000          ....
PP            RR       RR    IIIIII      000000          ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
                                        .SBTTL  PR.PAR - FIND PARTITION

                                ;+
                                ; PR.PAR - FIND PARTITION
                                ;
                                ; THIS ROUTINE IS CALLED TO FIND THE PROCESS PARTITION. IF THE PARTITION IS A
                                ; SYSTEM CONTROLLED PARTITION, A SUB-PARTITION PCB WILL BE CREATED.
                                ;
                                ; INPUTS:
                                ;       R0=LABEL BLOCK BUFFER ADDRESS
                                ;       $$PAR=PARTITION NAME (OPTIONAL)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       $PNAME=PARTITION NAME
                                ;       PCBA=PROCESS PCB ADDRESS
                                ;-
                                        .IF DF  R$$11M
000752  016767  000002G 000016' PR.PAR:  MOV     $$PAR+2,$PNAME+2 ; COPY PARTITION OVERRIDE NAME
000760  016767  000000G 000014'         MOV     $$PAR,$PNAME     ; ...
000766  001006                          BNE     10$              ; IF NE, USE OVERRIDE NAME
000770  016067  000004  000014'         MOV     L$BPAR(R0),$PNAME ; ELSE USE PARTITION FROM PROCESS FILE
000776  016067  000006  000016'         MOV     L$BPAR+2(R0),$PNAME+2 ; ...
001004                          10$:    SWSTK$  60$              ;* ENTER KERNEL MODE
001010  016705  000000G                 MOV     PARHD,R5         ;* GET POINTER TO FIRST PCB
001014  011505                  20$:    MOV     (R5),R5          ;* GET NEXT PCB
001016  001515                          BEQ     101$             ;* IF ZERO, END OF LIST
001020  026765  000014' 000004          CMP     $PNAME,P.NAM(R5) ;* LOOK FOR MATCH BETWEEN $PNAME
001026  001372                          BNE     20$              ;* AND PARTITION NAME FROM PCB
001030  026765  000016' 000006          CMP     $PNAME+2,P.NAM+2(R5) ;* ...
001036  001366                          BNE     20$              ;*      ...

                                        .IF DF  R$$MPL

                                        CALL    PR.SUB           ;* CREATE SUB-PARTITION

                                        .IFF

                                           .IF DF       M$$MGE

001040  032765  000040  000030          BIT     #PS.SYS,P.STAT(R5) ;* SYSTEM CONTROLLED PARTITION?
001046  001407                          BEQ     25$              ;* IF EQ, NO
001050  032767  00000uG 00000uG         BIT     #FL.CEX,$FLAGS   ;* IS THIS THE COMM EXEC?
001056  001111                          BNE     131$             ;* IF NE, YES
001060                                  CALL    PR.SUB           ;* CREATE SUB-PARTITION
001064  000462                          BR      40$              ;*
001066  032767  000000G 000000G 25$:    BIT     #FL.CEX,$FLAGS   ;* IS THIS THE COMM EXEC?
001074  001430                          BEQ     30$              ;* IF EQ, NO
001076  016501  000014                  MOV     P.REL(R5),R1     ;* GET PARTITION START IN BLOCKS
001102  010102                          MOV     R1,R2            ;* COPY IT
001104                                  ASL$    6,R1             ;* CONVERT TO BYTES
001120  026001  000010                  CMP     L$BSA(R0),R1     ;* WAS CEX BUILT FOR THIS PARTITION?
001124  001056                          BNE     111$             ;* IF NE, NO
001126  066502  000016                  ADD     P.BLKS(R5),R2    ;* ADD SIZE TO START ADDRESS
001132  022702  001000                  CMP     #1000,R2         ;* IS PARTITION WITHIN FIRST 16K?
```

PRIO CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-83	9-416	9-417	10-507						
ASR$	#5-93	9-396	9-409							
CACDF$	#5-60	5-67								
CALL	8-223	8-228	8-231	8-233	8-234	8-240	9-286	9-330	9-343	9-366
	9-387	10-480	10-501	11-599	13-770					
CALLR	#5-62	9-288	9-292	9-298	9-309	11-667				
CRASH	8-262									
DHBDF$	#5-62	5-66								
ECDDB$	#5-60	5-68								
LBLDF$	#5-62	5-64								
NHWDF$	#5-60	5-74								
PCBDF$	#5-62	5-65								
RESRG	#5-60	9-389								
RETC	#5-60									
RETURN	8-253	9-371	9-377	9-434	9-452	10-552	10-556	11-657	11-663	13-787
	13-794									
SAVRG	#5-60	9-384								
SOB	11-605									
SWSTK$	9-366	10-480								

FILEID**PROAL

```
                        .TITLE  PROERR - PROCESS RELATED ERRORS
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - COMMON "SET CEX" AND "SET PROCESS" ERROR MESSAGES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

Table of contents

5- 43 MACRO DEFINITIONS
6- 78 $NICAH - ALLOCATE XPT CACHE
7- 137 $ADJ - ALLOCATE ADJACENCY DATA BASE
8- 210 $ROL - ALLOCATE REACHIBILITY MATRIX
9- 312 $CKT - ALLOCATE CIRCUIT COUNTERS

PROXPT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES			
T$LLDS	000012	#5-67			
T$LLEN	000046	#5-67	9-327	9-352	
T$LOPR	000002	#5-67			
T$LTCL	000024	#5-67			
T$LTIM	000026	#5-67			
T$LTPR	000014	#5-67			
T$LTPS	000020	#5-67			
T$NAPL	000004	#5-67			
T$NFE	000000	#5-67			
T$NLEN	000010	#5-67			
T$NNUL	000002	#5-67			
T$NOPL	000006	#5-67			
T$NRNI	000042	#5-67			
T$NRPL	000005	#5-67			
T$NRUL	000007	#5-67			
T$NVR	000001	#5-67			
T$RPRI	000040	#5-67			
T$SVC	000034	#5-67			
T$T5	000030	#5-67			
T$T6	000032	#5-67			
Z.DSP	= ****** GX	6-112	8-286		
Z.LLN	= ****** GX	7-154	8-225		
$ADJ	000172 RG	#7-148			
$ALN18	= ****** GX	7-166	7-197	8-242	9-332
$CKT	001062 RG	#9-323			
$ERR9D	= ****** GX	7-168	7-199	8-244	9-334
$MAPX	= ****** GX	7-175	8-270	9-345	
$MUL	= ****** GX	6-105	7-159	9-328	
$NICAH	000000 RG	#6-92			
$PDVA	= ****** GX	6-96	7-152	8-223	
$ROL	000466 RG	#8-222			
$SAVAL	= ****** GX	6-92	7-148	8-222	9-323

PROXPT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 3

```
**FILE**ID**TMPACT
```

```
TTTTTTTTTT   MM       MM   PPPPPPPP       AAAAAA       CCCCCCCC   TTTTTTTTTT
TTTTTTTTTT   MM       MM   PPPPPPPP       AAAAAA       CCCCCCCC   TTTTTTTTTT
    TT       MMMM   MMMM   PP      PP    AA      AA   CC             TT
    TT       MMMM   MMMM   PP      PP    AA      AA   CC             TT
    TT       MM  MM  MM   PP      PP    AA      AA   CC             TT
    TT       MM  MM  MM   PP      PP    AA      AA   CC             TT
    TT       MM       MM   PPPPPPPP     AA      AA   CC             TT
    TT       MM       MM   PPPPPPPP     AA      AA   CC             TT
    TT       MM       MM   PP           AAAAAAAAAA   CC             TT
    TT       MM       MM   PP           AAAAAAAAAA   CC             TT
    TT       MM       MM   PP           AA      AA   CC             TT          ....
    TT       MM       MM   PP           AA      AA   CC             TT          ....
    TT       MM       MM   PP           AA      AA     CCCCCCCC     TT          ....
    TT       MM       MM   PP           AA      AA     CCCCCCCC     TT          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        ;+
                                        ; $ODD - ".ODD" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R2=DESTROYED
                                        ;-
000520                                  $ODD::
000520  032767  000001  000000G                 BIT     #1,$SIZE        ; IS SIZE CURRENTLY ODD?
000526  001010                                  BNE     10$             ; IF NE, YES - DO NOTHING
000530                                          CALL    $RQ2            ; REQUEST BINARY FILE
000534  103405                                  BCS     10$             ; IF CS, NO ACCESS
000536  112722  000000G                         MOVB    #.BYTE,(R2)+    ; STORE FUNCTION CODE
000542  105012                                  CLRB    (R2)            ; STORE ZERO VALUE
000544  005267  000000G                         INC     $SIZE           ; ADJUST DATA BASE SIZE
000550                                  10$:    RETURN
```

```
TMPACT - TEMPLATE PROCESSING AC MACRO V05.03b Saturday 29-Jun-85 00:38  Page 27-2
Symbol table

Size of work file: 12021 Words  ( 47 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:20.02
SY:TMPACT.V2,[132,134]TMPACT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPACT
```



```
TMPACT     CREATED BY  MACRO  ON 29-JUN-85 AT 00:38     PAGE 1
```

```
                                                        ;
000620  032767  000000G 000000G 20$:    BIT     #FL.PX1,$FLAGS  ; WAS DLC TRANSFERRED ?
000626  001405                         BEQ     30$             ; IF EQ, NO
000630  016704  000000G                MOV     $DLCPV,R4       ; GET DLC PDV ADDRESS
000634  001415                         BEQ     101$            ; IF ZERO, LOGIC ERROR
000636                                 CALL    XPRO            ; RELEASE DLC FROM PARTITION
000642  032767  000000G 000000G 30$:    BIT     #FL.PX2,$FLAGS  ; WAS DDM TRANSFERRED?
000650  001405                         BEQ     40$             ; IF EQ, NO
000652  016704  000000G                MOV     $DDMPV,R4       ; GET DDM PDV ADDRESS
000656  001405                         BEQ     111$            ; IF ZERO, LOGIC ERROR
000660                                 CALL    XPRO            ; RELEASE DDM FROM PARTITION
                                                        ;
000664  000261                 40$:    SEC                     ; RETURN WITH C-BIT SET
000666                                 RETURN
                                                        ;
                                        ; ERROR CONDITIONS
                                                        ;
000670                          101$:   CRASH                   ; NETLDR LOGIC ERROR
000672                          111$:   CRASH                   ; DITTO
```

```
                                        .SBTTL  ZDDS - CLEAR DDM LINE TABLE ADDRES WORD(S) AND KRB ADDRESS
                                ;+
                                ;
                                ; ZDDS - RE-SCAN SLT'S AND CLEAR DDM LINE TABLE ADDRESS WORD(S) AND KRB ADDRESS
                                ;
                                ; INPUTS:
                                ;       R5 - SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R3,R4,R5 PRESERVED
                                ;
                                ;-

002010  C   '  000000G         ZDDS:   JSR     R5,$SAVRG       ; SAVE R3,R4,R5
002014  017702  000000G                 MOV     @SLTMA,R2       ; GET SLT TABLE ADDRESS

002020  017701  000000G                 MOV     @SLTNM,R1       ; GET NUMBER OF SLT'S
002024  012200          10$:    MOV     (R2)+,R0        ; GET NEXT SLT
002026  126065  000002  000002          CMPB    L.DDM(R0),L.DDM(R5) ; SAME DEVICE AS OURS?
002034  001034                          BNE     20$             ; IF NE, NO
002036  126065  000012  000012          CMPB    L.CTL(R0),L.CTL(R5) ; SAME CONTROLLER?
002044  001030                          BNE     20$             ; IF NE, NO

002046  005060  000004                  CLR     L.DDS(R0)       ; ZERO DDM LINE TABLE ADDRESS
002052  005060  000016                  CLR     L.KRBA(R0)      ; ZERO KRB ADDRESS
002056  105060  000021                  CLRB    L.OWNR(R0)      ; CLEAR THE LINE OWNER
002062  116003  000014                  MOVB    L.NSTA(R0),R3   ; GET THE NUMBER OF STATIONS
002066  001411                          BEQ     15$             ; IF EQ, NONE
002070  010004                          MOV     R0,R4           ; POINT TO START OF TRIBUTARIES
002072  062704  000022                  ADD     #L.MPF,R4       ; ...
002076  105064  000003          13$:    CLRB    S.OWNR(R4)      ; CLEAR STATION OWNERS
002102  062704  000004                  ADD     #S.LEN,R4       ; NEXT STATION
002106                                  SOB     R3,13$          ; CLEAR ALL STATIONS

002112  126060  000002  000003  15$:    CMPB    L.DDM(R0),L.DLC(R0) ; COMBINATION DDM/DLC?
002120  001002                          BNE     20$             ; IF NE, NO
002122  005060  000010                  CLR     L.DLS(R0)       ; ZERO DLC LINE TABLE ADDRESS
002126                          20$:    SOB     R1,10$          ; SCAN ENTIRE SLT
002132                                  RETURN
```

```
                                        .SBTTL  XFLTTB - DE-ALLOCATE FILTER CONTROL BLOCKS

                                ;+
                                ;
                                ; XFLTTB - DE-ALLOCATE THE FILTER CONTROL BLOCK TABLE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUPUTS:
                                ;       ALL FILTER CONTROL BLOCKS ARE DE-ALLOCATED
                                ;
                                ;-

003272  016703  000000G         XFLTTB: MOV     FILHD,R3        ; POINT AT START OF FILTER TABLE LIST

003276                                  SWSTK$  30$             ;; ENTER SYSTEM STATE
003302                                  SAVMAP                  ;; SAVE CURRENT MAPPING
003306  011300                  10$:    MOV     (R3),R0         ;; GET THE NEXT FILTER BLOCK IN LIST
003310  001414                          BEQ     20$             ;; IF EQ, END OF LIST
003312  010046                          MOV     R0,-(SP)        ;; SET BLOCK ADDRESS
003314                                  CALL    $CEACX          ;; GET ACCESS TO BLOCK
003320  012604                          MOV     (SP)+,R4        ;; GET MAPPED ADDRESS OF BLOCK
003322  011413                          MOV     (R4),(R3)       ;; GET THE NEXT FILTER BLOCK IN LIST
003324  016401  000002                  MOV     F.CLS(R4),R1    ;; GET LENGTH OF BLOCK (IN LOWER 6 BITS)
003330  042701  177700                  BIC     #^C<77>,R1      ;; ISOLATE LENGTH
003334                                  CALL    $XDEAC          ;; DE-ALLOCATE THE FILTER BLOCK
003340  000762                          BR      10$             ;; AND GO AGAIN
003342                          20$:    RESMAP                  ;; RESTORE PREVIOUS MAPPING
003346                                  RETURN                  ;; EXIT SYSTEM STATE

003350                          30$:    RETURN
```

Symbol table

```
E$NCS   000036          FF.MON= 000004          FS.WLB= 001000          LF.UNL= 020000          M$3103= 000003
E$NIC   000044          FF.MSK= 000077          FS.XKL= 002000          LF.X2P= 000000          M.LGTH= ****** GX
E$NLEN  000050          FF.PRT= 000200          FS.XOF= 010000          LGDDB = ****** GX       NETPF = ****** GX
E$NLLA  000012          FF.QUL= 000370          FS.XON= 007000          LGPDV = ****** GX       NF$BLK= 000100
E$NLNK  000000          FF.REM= 100000          FS.ZER= 002000          LGSTT = ****** GX       NF$DMO= 000010
E$NML   000040          FILHD = ****** GX       F$$LVL= 000001          LLCAL = ****** GX       NF$MOU= 000040
E$NMR   000024          FLT     003352R         F.ADD   000016                                  NF$RST= 000002
E$NMS   000030          FLT16   003352R         F.CEV   000030          LN.CLO= 000000          NF$SCN= 000020
E$NNOD  000002          FL.PX1= ****** GX       F.CIR   000024          LN.DUM= 000005          NF$SHU= 000004
E$NOD   000010          FL.PX2= ****** GX       F.CLS   000002          LN.LOA= 000004          NF$TIM= 000200
E$NRT   000042          FMASK = ****** GX       F.EVT   000004          LN.LOO= 000003          NSITAB  000126R     002
E$NRTP  000005          FMKTMP  000002R     002 F.FLG   000014          LN.OAU= 000003          NS0   = ****** GX
E$NSEG  000010          FNDADD= ****** GX       F.LEN   000016          LN.OFF= 000001          NS1   = ****** GX
E$NTIM  000046          FR$BCC= 000007          F.LIN   000020          LN.ON = 000000          NS2   = ****** GX
E$NUSE  000004          FR$CCF= 000001          F.LNK   000000          LN.OOP= 000004          NS3   = ****** GX
E$PORT  000014          FR$CDF= 000002          F.MOD   000022          LN.OPE= 000001          NS4   = ****** GX
E$PRM   000002          FR$DAO= 000011          F.NXT   000000          LN.REF= 000002          NS5   = ****** GX
E$STAT  000006          FR$EXC= 000000          F.REM   000026          LN.SER= 000002          NS6   = ****** GX
E$STRT  000006          FR$FRM= 000010          F.SEV   000004          LN.STA= 000017          NS7   = ****** GX
E$TCB   000004          FR$FTL= 000005          F.USE   000002          LN.SUB= 000360          NTLPT = ****** GX
E$$XPR= 000000          FR$OPN= 000004          F.VAR   000003          LN.TRI= 000006          N$ACQ   000000
E.CTL   000020          FR$RFD= 000006          GETIND= ****** GX       L$$ASG= 000000          N$ACTL  000032
E.DATA  000046          FR$SBU= 000012          GETPDV= ****** GX       L$$DRV= 000000          N$ADJ1  000072
E.EVT   000002          FR$SHO= 000003          G$$TPP= 000000          L$$P11= 000001          N$ADJ2  000074
E.LCN   000042          FR$UBU= 000013          G$$TSS= 000000          L$$11R= 000000          N$CACH  000062
E.LEN   000216          FR$UPT= 000014          G$$TTK= 000000          L.COST  000015          N$CIR   000034
E.LIN   000024          FS.AST= 000000          G$$WRD= 000000          L.CTL   000012          N$CRC   000120
E.LNK   000000          FS.CIB= 002000          ICBLEN  001720R         L.CVA   177776          N$DLA   000020
E.MOD   000036          FS.CRA= 001000          ICBSC   001662R         L.DDM   000002          N$DLY   000014
E.NOD   000034          FS.DIS= 013000          ICBSZ   000006R     002 L.DDS   000004          N$ELEN  000054
E.PDV   000021          FS.DVC= 001000          INIFL = 000001          L.DLC   000003          N$ENC   000042
E.PORT  000040          FS.ENB= 012000          I$$RAR= 000000          L.DLM   000006          N$ERRC  000022
E.PRM   000026          FS.EXI= 001000          I$$RDN= 000000          L.DLS   000010          N$FLG   000005
E.PVC   000044          FS.GET= 006000          KMCB    000116R     002 L.FLG   000000          N$FNC   000006
E.SIZ   000022          FS.HLT= 000000          KSAR5 = ****** GX       L.KRBA  000016          N$GENQ  000052
E.TIME  000004          FS.INI= 000000          K$$CNT= 177546          L.LEN = 000022          N$GTM   000015
FC.CCP= 000020          FS.KIL= 000000          K$$CSR= 177546          L.MPF   000022          N$HC1   000052
FC.CTL= 000006          FS.LCL= 100000          K$$LDC= 000000          L.NMST  000020          N$HC2   000056
FC.KCP= 000016          FS.LTM= 001000          K$$TPS= 000074          L.NSTA  000014          N$HIGH  000033
FC.KIL= 000004          FS.MNT= 004000          K.CSR   000002          L.OWNR  000021          N$LLT   000026
FC.MAN= 000024          FS.MSN= 014000          K.PRI   000000          L.UNT   000013          N$LLTM  000024
FC.MLD= 000026          FS.REA= 001000          K.VCT   000001          MO$SAC= 000016          N$LVC   000036
FC.PCT= 000030          FS.RET= 000000          K6TMP   000000R     002 MO$SPR= 000012          N$LV1   000002
FC.PWR= 000022          FS.REZ= 003000          LD$LP = 000000          MO$SSV= 000014          N$LV2   000010
FC.RCE= 000002          FS.RLB= 002000          LF.ACT= 100000          MO$25A= 000006          N$MBXQ  000050
FC.RCP= 000014          FS.RNG= 011000          LF.BRO= 000400          MO$25P= 000002          N$MHC1  000036
FC.TIM= 000010          FS.RST= 000000          LF.BWT= 000007          MO$25S= 000004          N$MHC2  000044
FC.XCP= 000012          FS.RTN= 001000          LF.ENA= 002000          MO$29S= 000010          N$PLD   000016
FC.XME= 000000          FS.SET= 005000          LF.LPB= 001000          M$HIGH= 000003          N$PLLT  000030
FE.CEX= ****** GX       FS.SFC= 005000          LF.MDC= 000100          M$$CRB= 000124          N$PRI   000076
FF.ADD= 000020          FS.SFR= 006000          LF.MFL= 004000          M$$CRX= 000000          N$ROA1  000022
FF.CIR= 000040          FS.SFS= 004000          LF.MTP= 000020          M$$FCS= 000000          N$ROA2  000030
FF.CON= 000001          FS.SPW= 040000          LF.PAC= 000200          M$$MGE= 000000          N$RTMX  000014
FF.FIL= 000002          FS.STM= 000000          LF.RDY= 040000          M$$NET= 000000          N$RTM1  000014
FF.HST= 040000          FS.STP= 002000          LF.REA= 010000          M$$OVR= 000000          N$RTM2  000015
FF.LIN= 000010          FS.STR= 001000          LF.SER= 000040          M$3100= 000000          N$RT1   000000
FF.MOD= 000100          FS.TRM= 003000          LF.TIM= 000010          M$3101= 000001          N$RT2   000006
                                                                        M$3102= 000002
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 11

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
$XSNA	= ****** GX	19-1164	19-1166							
$$PDIN	= ****** GX	9-342	*12-488	*19-1115						
$$$ARG	= 000004	#8-293	8-293	8-293	8-293	8-293	8-293	8-293	8-293	8-293
		8-293	8-293							
.CXALL	= ****** GX	19-1196	19-1273							
.CXLEN	= ****** GX	19-1197	19-1274							
.CXPCB	= ****** GX	19-1204								
.CXUNL	= ****** GX	27-1582	28-1730	*28-1732	28-1736					
.DEA16	004166 R	28-1676	28-1694	#29-1803						
.NAME	000146 R	#7-191								
.NTLPT	000150 R	#7-192								
.PCB	= ****** GX	19-1306								
.UMRBA	= ****** GX	19-1327								
..PDV	000152 R	#7-202	*8-272	*11-441	*14-607	14-630	*14-644	14-652	29-1818	

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 12

Table of contents

5- 55 MACROS
6- 77 LOCAL DATA
7- 87 $XALOC - ALLOCATE CORE BUFFER IN EXTENDED POOL
8- 178 $XDEAC - DEALLOCATE CORE BUFFER IN EXTENDED POOL
9- 300 $LINKX - LINK/UNLINK BLOCK IN EXTENDED POOL
10- 341 $CEACX - ACCESS BLOCK IN EXTENDED POOL

POLAL CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES								
ALOCB	= ****** GX	7-131								
DEACB	= ****** GX	8-215								
KSAR5	= ****** GX	7-119	7-172	8-233	8-240	8-257	8-258	8-260	8-265	8-271
		8-273	8-280	8-281	8-283	8-287	8-294	10-394	10-395	
M$$MGE	= 000000	7-113	8-204	10-361						
R$$EIS	= ******	10-379								
R$$11M	= 000000	7-129	8-214	10-369						
XAVL	= ****** GX	7-123	7-173	8-245	8-289					
XAVLL	000002 R	#6-84	7-120	7-173	8-241	8-289				
XBIAS	= ****** GX	10-394								
$CEACX	000470 RG	7-154	7-168	8-238	8-250	8-267	9-328	9-333	#10-360	
$LINKX	000424 RG	#9-324								
$XALOC	000006 RG	#7-112								
$XDEAC	000134 RG	#8-202								

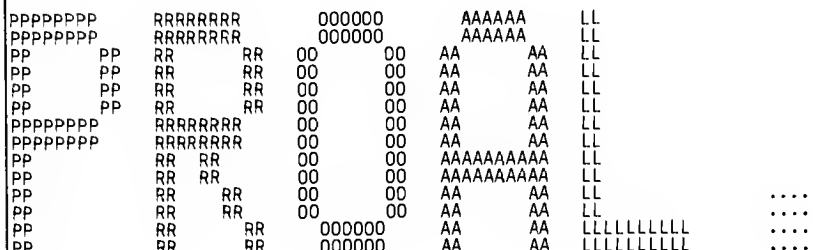
POLAL CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

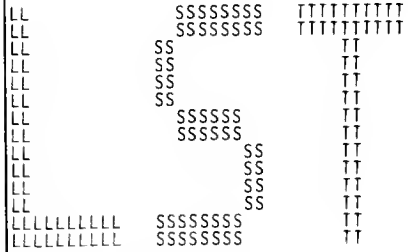
Table of contents


```
    5-   55    MACRO DEFINITIONS
    6-  103    LOCAL SYMBOL DEFINTIONS
    7-  135    LOCAL DATA
    8-  196    $PRIO - PROCESS I/O
    9-  264    PR.LBL - READ LABEL BLOCK
   10-  456    PR.PAR - FIND PARTITION
   11-  576    PR.SUB - CREATE SUB-PARTITION
   13-  742    PR.XFR - READ PROCESS INTO CORE
```

```
                                                .IF DF  X$$MDB

                                        BLO     151$            ;* IF LO, NO

                                                .IFF    ; DF X$$MDB

001136  103007                          BHIS    30$             ;* IF HIS, YES
001140  032777  000000G 000000G         BIT     #FE.EXV,@FMASK  ;* IS 20K EXEC SUPPORTED?
001146  001465                          BEQ     151$            ;* IF EQ, NO
001150  022702  001200                  CMP     #1200,R2        ;* IS PARTITION WITHIN FIRST 20K?
001154  103462                          BLO     151$            ;* IF LO, NO

                                                .ENDC   ; DF X$$MBD

001156  026765  000000' 000016  30$:    CMP     BLOX,P.BLKS(R5) ;* IS PARTITION BIG ENOUGH?
001164  101042                          BHI     121$            ;* IF HI, NO

                                           .IFF ; DF M$$MGE

                                        CMP     L$BSA(R0),P.REL(R5) ;* WAS PROCESS BUILT FOR THIS PARTITION?
                                        BNE     111$            ;* IF NE, NO
                                        CMP     BYTES,P.SIZE(R5) ;* IS PARTITION BIG ENOUGH?
                                        BHI     121$            ;* IF HI, NO

                                           .ENDC        ; DF M$$MGE

001166  032765  000200  000030          BIT     #PS.COM,P.STAT(R5) ;* COMMON PARTITION?
001174  001442                          BEQ     131$            ;* IF EQ, NO
001176  105765  000025                  TSTB    P.BUSY+1(R5)    ;* IS PARTITION BUSY?
001202  001043                          BNE     141$            ;* IF NE, YES
001204  016504  000012                  MOV     P.MAIN(R5),R4   ;* GET MAIN PARTITION PCB ADDRESS
001210  105764  000025                  TSTB    P.BUSY+1(R4)    ;* IS MAIN PARTITION BUSY?
001214  001036                          BNE     141$            ;* IF NE, YES
001216  156565  000024  000025          BISB    P.BUSY(R5),P.BUSY+1(R5) ;* SET PARTITION BUSY
001224  156564  000024  000025          BISB    P.BUSY(R5),P.BUSY+1(R4) ;* SET MAIN PARTITION BUSY

                                        .ENDC   ; DF R$$MPL

001232  010567  000012'        40$:    MOV     R5,PCBA         ;* SET PCB ADDRESS
001236                           50$:    RETURN                  ;* BACK TO USER MODE
001240  005767  000010'          60$:    TST     ERROR           ; DID ERROR OCCUR IN KERNEL MODE? (CLEARS C-BIT)
001244  001401                          BEQ     80$             ; IF ZERO, NO
001246  000261                  70$:    SEC                     ; INDICATE FAILURE
001250                           80$:    RETURN

                                 ;
                                 ; ERROR CONDITIONS
                                 ;
001252  012767  000022  000010' 101$:    MOV     #ERR18,ERROR    ;* PARTITION NOT IN SYSTEM
001260  000766                          BR      50$             ;*
001262  012767  000024  000010' 111$:    MOV     #ERR20,ERROR    ;* PROCESS/PARTITION BASE ADDRESS MISMATCH
001270  000762                          BR      50$             ;*
001272  012767  000026  000010' 121$:    MOV     #ERR22,ERROR    ;* PARTITION TOO SMALL
001300  000756                          BR      50$             ;*
001302  012767  000030  000010' 131$:    MOV     #ERR24,ERROR    ;* NOT COMMON PARTITION
001310  000752                          BR      50$             ;*
001312  012767  000032  000010' 141$:    MOV     #ERR26,ERROR    ;* PARTITION BUSY
```

FILEID**PROAL


PPPPPPPP RRRRRRRR 000000 AAAAAA LL
PPPPPPPP RRRRRRRR 000000 AAAAAA LL
PP PP RR RR 00 00 AA AA LL
PP PP RR RR 00 00 AA AA LL
PP PP RR RR 00 00 AA AA LL
PP PP RR RR 00 00 AA AA LL
PPPPPPPP RRRRRRRR 00 00 AA AA LL
PPPPPPPP RRRRRRRR 00 00 AA AA LL
PP RR RR 00 00 AAAAAAAAAA LL
PP RR RR 00 00 AAAAAAAAAA LL
PP RR RR 00 00 AA AA LL ....
PP RR RR 00 00 AA AA LL ....
PP RR RR 000000 AA AA LLLLLLLLLL ....
PP RR RR 000000 AA AA LLLLLLLLLL ....



```
                        .SBTTL  MACRO DEFINITIONS
;***
; LIBRARY MACROS
;***
                        .MCALL  NTLER$
```

```
                        .TITLE  PROXPT - XPT PROCESS INITIALIZER
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - "INIT XPT" ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

PROXPT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ADJDF$	#5-45	5-66								
ASR$	#5-46	7-164	7-196	8-240	9-331					
BIAS	#5-45	8-253	9-339							
BIAS5	#5-56	6-117	8-291							
CACDF$	#5-45	5-71								
CALL	6-92	6-98	6-105	7-148	7-159	7-166	7-168	7-174	7-175	7-197
	7-199	8-222	8-242	8-244	8-266	8-270	9-323	9-328	9-332	9-334
	9-343	9-345								
CTRDF$	#5-45	5-67								
DHBDF$	#5-45	5-70								
ECDDB$	#5-45	5-68								
EMSG$	#5-46	7-168	7-199	8-244	9-334					
MAP	#5-61	6-112	8-286							
RESMAP	#5-52	6-128	8-302							
RESRG	#5-45									
RETURN	6-129	6-134	7-169	7-189	7-200	7-207	8-246	8-303	8-309	9-359
SAVMAP	#5-48	6-109	8-282							
SAVRG	#5-45									
SOB	6-126	8-300	9-355	9-357						
SWSTK$	6-98	7-174	8-266	9-343						
XPDDB$	#5-45	5-69								

FILEID**SYMBL

```
                        .TITLE  IMPACT - TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 1
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                                ;+
                                ; $PRI - ".PRI" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000552                          $PRI1::                         ; ENTRY FOR DECREMENTING DUP PRIORITY
000552  005267  000000G                INC     $$PRI           ; UPDATE PRIORITY (COMPLEMENT OF PRIORITY)
000556                          $PRI::
000556                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000562                                  CALL    $RQ1            ; REQUEST BINARY BUFFER ACCESS
000566  103405                          BCS     10$             ; IF CS, ACCESS DENIED
000570  112712  000000G                MOVB    #.PRI,(R2)      ; STORE FUNCTION CODE
000574  062767  000002  000000G        ADD     #2,$SIZE        ; ADJUST LINE TABLE SIZE
000602                          10$:    RETURN
```

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
FL.DDM	= ******	GX	27-662								
FL.DLC	= ******	GX	27-662								
FL.MUX	= ******	GX	13-266								
FMT2	000422	R	5-74	5-75	5-76	5-77	5-78	5-79	5-80	5-81	#5-90
FM.2	= 000000		#5-74	#5-75	#5-76	#5-77	#5-78	#5-79	#5-80	#5-81	
F1.LFL	= ******	GX	14-296								
F1.MPL	= ******	GX	14-301								
F2.BYT	= ******	GX	23-564								
F2.FIL	= ******	GX	26-643								
F2.NEG	= ******	GX	23-568	23-572	23-577						
F2.SUB	= ******	GX	24-598	24-604							
LF.ACT	= 100000		#5-61								
LF.BRO	= 000400		#5-61								
LF.BWT	= 000007		#5-61								
LF.ENA	= 002000		#5-61								
LF.LPB	= 001000		#5-61								
LF.MDC	= 000100		#5-61								
LF.MFL	= 004000		#5-61								
LF.MTP	= 000020		#5-61								
LF.PAC	= 000200		#5-61								
LF.RDY	= 040000		#5-61								
LF.REA	= 010000		#5-61								
LF.SER	= 000040		#5-61								
LF.TIM	= 000010		#5-61								
LF.UNL	= 020000		#5-61								
LF.X2P	= 000000		#5-61								
LN.CLO	= 000000		#5-61								
LN.DUM	= 000005		#5-61								
LN.LOA	= 000004		#5-61								
LN.LOO	= 000003		#5-61								
LN.OAU	= 000003		#5-61								
LN.OFF	= 000001		#5-61								
LN.ON	= 000000		#5-61								
LN.OOP	= 000004		#5-61								
LN.OPE	= 000001		#5-61								
LN.REF	= 000002		#5-61								
LN.SER	= 000002		#5-61								
LN.STA	= 000017		#5-61								
LN.SUB	= 000360		#5-61								
LN.TRI	= 000006		#5-61								
L.COST	000015		#5-61								
L.CTL	000012		#5-61	8-171	8-178						
L.CVA	177776		#5-61								
L.DDM	000002		#5-61								
L.DDS	000004		#5-61								
L.DLC	000003		#5-61								
L.DLM	000006		#5-61								
L.DLS	000010		#5-61								
L.FLG	000000		#5-61								
L.KRBA	000016		#5-61								
L.LEN	= 000022		#5-61								
L.MPF	000022		#5-61								

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 2

```
                                        .SBTTL  $CPRO - CLEAR PROCESS

                                ;+
                                ; $CPRO - CLEAR PROCESS
                                ;
                                ; THIS ROUTINE IS CALLED TO REMOVE A PROCESS FROM CORE. THE MARK-FOR-LOAD
                                ; BIT IN THE PDV IS ALWAYS CLEARED.
                                ;
                                ; INPUTS:
                                ;       $PDVA=PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000674  016704  000000G         $CPRO:: MOV     $PDVA,R4        ; GET PDV ADDRESS
000700  032764  000000G 000000G         BIT     #ZF.INI,Z.FLG(R4) ; DOES PROCESS NEED TERMINATION DISPATCH ?
000706  001420                          BEQ     15$             ; IF NO - BRANCH
000710  016467  000000G 000000G         MOV     Z.PCB(R4),$PCB  ; SAVE PCB ADDRESS
000716                                  CALL    DQCLK           ; DEQUEUE 100MSEC CLOCK BLOCK

000722  012767  003030  000000G         MOV     #FS.TRM!FC.PCT,$$PDIN ; SET UP FUNCTION CODE
000730  010401                          MOV     R4,R1           ; COPY PDV ADDRESS
000732                                  CALL    $DSINI          ; CALL PROCESS
000736  005064  000000G                 CLR     Z.PCB(R4)       ; DISABLE TIMER SERVICE
000742  000241                  10$:    CLC                     ; CLR C-BIT
000744                                  CALL    XDCB            ; UNLOAD DRIVER
000750                          15$:    CALL    XPRO            ; RELEASE PROCESS FROM MEMORY
000754  142764  000000G 000000G         BICB    #ZF.MFL,Z.FLG(R4) ; CLEAR THE MARK-FOR-LOAD FLAG
000762  000241                          CLC                     ; CLR C-BIT
000764                                  RETURN
```

```
                                        .SBTTL  $CCEX - CLEAR CEX

                                ;+
                                ; $CCEX - CLEAR CEX
                                ;
                                ; THIS ROUTINE IS CALLED TO REMOVE ALL COMM EXEC RELATED PIECES FROM CORE.
                                ; IT IS ASSUMED THAT THE CALLER IS USING HIS/HER GOOD JUDGEMENT BECAUSE NO
                                ; CHECKING OF ANY KIND IS DONE TO SEE IF THIS FUNCTION IS SAFE.
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       EVERYTHING IS REMOVED FROM CORE
                                ;-
002134                          $CCEX::
002134  032777  000000G 000000G         BIT     #FE.CEX,@FMASK  ; IS COMM EXEC LOADED ?
002142  001002                          BNE     1$              ; IF NE, YES
002144  000167  000530                  JMP     80$             ; ELSE, ERROR RECOVERY

002150                          1$:
002150                                  SWSTK$  9$              ;; ENTER KERNEL MODE
002154  005077  000000G                 CLR     @NETPF          ;; CLEAR POWERFAIL FLAG

002160  016700  000000G         2$:     MOV     CLKHD,R0        ;; POINT AT CLOCK QUEUE
002164                                  MTPS    #PR7            ;;; DISABLE INTERRUPTS
002172  010001                  3$:     MOV     R0,R1           ;;; COPY CURRENT BLOCK
002174  011100                          MOV     (R1),R0         ;;; GET NEXT BLOCK
002176  001405                          BEQ     6$              ;;; IF ZERO, END OF LIST
002200  026700  000000G                 CMP     T1SCL,R0        ;;; 1 SEC TIMER ENTRY ?
002204  001372                          BNE     3$              ;;; IF NE, NO
002206  011011                  4$:     MOV     (R0),(R1)       ;;; REMOVE ENTRY FROM QUEUE
002210  000770                          BR      3$              ;;; GO GET NEXT
002212                          6$:     MTPS    #0              ;;; RE-ENABLE INTERRUPTS
002216                                  RETURN                  ;; BACK TO USER MODE

002220  017704  000000G         9$:     MOV     @SLTMA,R4       ; GET SLT TABLE ADDRESS
002224  017746  000000G                 MOV     @SLTNM,-(SP)    ; AND COUNT OF SLT'S
002230  001415                          BEQ     30$             ; IF ZERO, NO WORK HERE

002232  012405                  10$:    MOV     (R4)+,R5        ; GET NEXT SLT
002234  032765  040000  000000         BIT     #LF.RDY,L.FLG(R5) ; IS THE LINE READY?
002242  001406                          BEQ     20$             ; IF EQ, NO
002244  010567  000000G                 MOV     R5,$SLTA        ; SAVE SLT ADDRESS
002250  010446                          MOV     R4,-(SP)        ; SAVE POINTER IN SLT MAPPING TABLE
002252                                  CALL    XLIN            ; UNLOAD THE LINE
002256  012604                          MOV     (SP)+,R4        ; RESTORE MAPPING INTO SLT TABLE
002260  005316                  20$:    DEC     (SP)            ; LOOP <$SLTNM> TIMES
002262  001363                          BNE     10$             ;
002264  017705  000000G         30$:    MOV     @PDVTA,R5       ; GET PDV TABLE ADDRESS

002270  017716  000000G                 MOV     @PDVNM,(SP)     ; AND COUNT OF PDV'S
002274  001435                          BEQ     52$             ; IF ZERO, NO WORK HERE
002276  012504                  40$:    MOV     (R5)+,R4        ; GET PDV ADDRESS
002300  001431                          BEQ     50$             ; ALLOW EMPTY SLOTS
```

```
                                        .SBTTL  FLT16 - FIND A 16 BIT ALLOCATION LINE TABLE

                                ;+
                                ; FLT16 - FIND A 16 BIT ALLOCATION LINE TABLE
                                ;
                                ; INPUTS:
                                ;       R3=LINE TABLE ADDRESS
                                ;       R4=PDV INDEX OF DDM/DLC/LLC PROCESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2,R3=DESTROYED
                                ;       UBIAS,UBA=UNLOAD BLOCK ADDRESS
                                ;       BLOCK=LOCAL COPY OF UNLOAD BLOCK
                                ;-

003352                          FLT16:
003352  005067  000012'         FLT:    CLR     UBA             ; INDICATE NO UNLOAD BLOCK FOUND YET

                                        .IF DF  M$$MGE
003356  017700  000000G                 MOV     @NTLPT,R0       ; GET NETLDR DATA BLOCK BIAS

                                        .IF DF  R$$11M
003362                                  SWSTK$  100$            ;; ENTER KERNEL MODE
                                        .ENDC   ; DF R$$11M

003366                                  CALL    $MAPX           ;; MAP TO THE DATA BLOCK
003372  016211  000000G                 MOV     .CXUNL(R2),(R1) ;; AND TO THE FIRST UNLOAD BLOCK
                                        .IFF    ; DF M$$MGE
                                        MOV     @NTLPT,R2       ; GET NETLDR DATA BLOCK ADDRESS
                                        MOV     .CXUNL(R2),R2   ; AND FIRST UNLOAD BLOCK ADDRESS
                                        .ENDC   ; DF M$$MGE

003376  001467                          BEQ     100$            ;; IF ZERO, NO UNLOAD BLOCKS
003400  005722                  10$:    TST     (R2)+           ;; SKIP POINTER TO NEXT UNLOAD BLOCK
003402  005722                  20$:    TST     (R2)+           ;; POINT PAST THE PDV INDEX
003404  126204  177776                  CMPB    -2(R2),R4       ;; IS THIS THE RIGHT PDV ?
003410  001013                          BNE     30$             ;; IF NE, NO
003412  022203                          CMP     (R2)+,R3        ;; IS THIS OUR LINE TABLE?
003414  001442                          BEQ     80$             ;; IF EQ, YES
003416  101011                          BHI     40$             ;; IF HI, NO
003420  011246                          MOV     (R2),-(SP)      ;; EXTRACT SIZE FIELD
003422  042716  000003                  BIC     #3,(SP)         ;;     ...
003426  066216  177776                  ADD     -2(R2),(SP)     ;; ADD TABLE ENTRY
003432  022603                          CMP     (SP)+,R3        ;; IS THIS OUR LINE TABLE?
003434  101032                          BHI     80$             ;; IF HI, YES
003436  000401                          BR      40$             ;; ELSE, SKIP ANY SUB-ALLOCATIONS
003440  005722                  30$:    TST     (R2)+           ;; SKIP THE LINE TABLE ADDRESS
003442  032722  000001          40$:    BIT     #1,(R2)+        ;; ANY SUB-ALLOCATIONS FOR THIS ENTRY?
003446  001410                          BEQ     50$             ;; IF EQ, NO
003450  005046                          CLR     -(SP)           ;; GET NUMBER OF ALLOCATIONS
003452  152216                          BISB    (R2)+,(SP)      ;;     ...
003454  005046                          CLR     -(SP)           ;; AND NUMBER OF LIBRARIES
003456  152216                          BISB    (R2)+,(SP)      ;;     ...
003460  062616                          ADD     (SP)+,(SP)      ;; ADD THEM TOGETHER
003462  006316                          ASL     (SP)            ;; MULTIPLY BY 4
003464  006316                          ASL     (SP)            ;;     ...
003466  062602                          ADD     (SP)+,R2        ;; GIVING NEXT TABLE ENTRY
```

Symbol table

```
N$SLA   000016          RE$OPR= 000000          SF.SER= 000001          T$LTIM  000026          T2.SPN= 000004
N$SNOD  000012          RE$RCV= 000001          SF.SVC= 000002          T$LTPR  000014          T2.STP= 000020
N$TCTL  000112          RE$SED= 000011          SF.TER= 001000          T$LTPS  000020          T2.WFR= 000001
N$TIM   000004          RE$SKW= 000006          SF.UNL= 000040          T$NAPL  000004          T3.ACP= 100000
N$TLC   000100          RE$STA= 000002          SLTMA = ****** GX       T$NFE   000000          T3.CAL= 000100
N$TNC   000114          RE$SUM= 000003          SLTNM = ****** GX       T$NLEN  000010          T3.CLI= 001000
N$TRC   000106          RE$SYN= 000000          SV$DUM= 000001          T$NNUL  000002          T3.GFL= 000010
N$TTCB  000110          RE$TME= 000021          SV$LOA= 000000          T$NOPL  000006          T3.MCR= 004000
N$VCB   000010          RE$TMO= 000000          S$$WRG= 000000          T$NRNI  000042          T3.NET= 000020
N$VER   000066          RE$TMR= 000020          S$$YSZ= 007600          T$NRPL  000005          T3.NSD= 000200
N$XLEN  000124          RE$UPT= 000002          S.ADD   000004          T$NRUL  000007          T3.PMD= 040000
N$$ACC= 000001          RE$URE= 000003          S.CIR   000016          T$NVR   000001          T3.PRV= 010000
N$$BUF= 000001          RE$VER= 000005          S.COST  000001          T$RPRI  000040          T3.REM= 020000
N$$LDV= 000001          RE$VRQ= 000015          S.DEV   000020          T$SVC   000034          T3.ROV= 000040
N$$MCP= 000001          RF.LOO= 100000          S.DIA   000046          T$T5    000030          T3.RST= 000400
N$$MLL= 000001          RT$INI= 000002          S.DPA   000032          T$T6    000032          T3.SLV= 002000
N$$MOV= 000010          RT$OFF= 000001          S.DPC   000036          T$$KMG= 000000          T3.SWS= 000002
N$$NCT= 000001          RT$ON = 000000          S.DUM   000050          T$$MIN= 000000          UBA     000012R   002
N$$PEM= 000001          R$R0   = 000002         S.FLAG  000002          T.ACTL  000052          UBIAS   000010R   002
OBJHD = ****** GX       R$R1   = 000004         S.FLG   000000          T.ASTL  000016          UPTR    000114R   002
OF.PRO= 000040          R$R2   = 000006         S.HAD   000010          T.ATT   000062          URMTMP  000004R   002
OF.RLU= 000100          R$R3   = 000010         S.HST   000006          T.CPCB  000004          VECT    001412RG
OF.SMC= 000200          R$R4   = 000012         S.LEN   000004          T.DPRI  000040          V$$CTR= 001000
OP$INI= 000000          R$R5   = 000014         S.LENG  000052          T.EFLG  000022          XALITB  003142R
OP$TER= 000001          R$$DER= 000000          S.LNK   000000          T.IOC   000003          XDCB    003046R
O.FLG   000003          R$$K11= 000001          S.LOA   000044          T.LBN   000041          XFILTB  004522R
O.LEN   000012          R$$SND= 000000          S.NMST  000002          T.LDV   000044          XFLTTB  003272R
O.LNK   000000          R$$11M= 000000          S.OWNR  000003          T.LNK   000000          XF.CAN= 000200
O.MXC   000004          R.ADD   000010          S.PSS   000022          T.MXSZ  000050          XF.DDM= 000001
O.NAM   000006          R.FLAG  000012          S.SEC   000040          T.NAM   000006          XF.DLC= 000002
O.TYP   000002          R.LEN   000014          S.TER   000042          T.OFF   000066          XF.LLC= 000004
PDVID = ****** GX       R.LNK   000000          TS.BLK= 171700          T.PCB   000046          XKRB  = ****** GX
PDVNM = ****** GX       R.NAM   000C02          TS.CKP= 000200          T.PRI   000002          XLIN    001106R
PDVTA = ****** GX       SCAN  = ****** GX       TS.CKR= 000100          T.RCVL  000012          XLT     003564R
PH$HDE= 000004          SC$OFF= 000001          TS.EXE= 100000          T.RRFL  000072          XLT16   003560R
PH$LOC= 000002          SC$ON = 000000          TS.HLD= 002000          T.SAST  000054          XOBJTB  003124R
PH$MTS= 000003          SC$RST= 000003          TS.MSG= 020000          T.SRCT  000071          XPRO    000022R
PH$UMP= 000000          SC$SHU= 000002          TS.NRP= 010000          T.STAT  000032          XREMTB  003242R
PH$WCS= 000001          SDBLH = ****** GX       TS.OUT= 000400          T.ST2   000034          XSERTB  004476R
PR7   = ****** GX       SF.ACT= 000200          TS.RDN= 040000          T.ST3   000036          X$$DBT= 000000
PS    = ****** GX       SF.CIR= 000010          TS.RUN= 004000          T.TCBL  000030          X.FLAG  000003
P$$P45= 000000          SF.DEV= 000020          TS.STP= 001000          T.TID   000057          X.ID    000002
P$$WRD= 000000          SF.DIA= 004000          T$FLAG  000044          T.TKSZ  000060          X.LINK  000000
P.REL = ****** GX       SF.DPA= 000100          T$LIF   000013          T.UCB   000026          X.RTMR  000006
Q$$OPT= 000010          SF.DPC= 000200          T$LIFL  000013          T1SCL = ****** GX       X.TMR   000004
RDBLH = ****** GX       SF.DUM= 010000          T$LIFO  000013          T100C = ****** GX       ZDDS    002010R
RE$ADC= 000004          SF.ENA= 000100          T$LIFS  000013          T100Q = ****** GX       ZF.INI= ****** GX
RE$ADF= 000017          SF.HAD= 000004          T$LIN   000000          T2.ABO= 000100          ZF.LLC= ****** GX
RE$ADR= 000007          SF.HST= 000002          T$LIPS  000006          T2.AST= 100000          ZF.MFL= ****** GX
RE$BLK= 000010          SF.LOA= 002000          T$LLD   000012          T2.CAF= 000400          ZF.PSE= ****** GX
RE$CAF= 000014          SF.LPB= 000004          T$LLDC  000045          T2.CHK= 020000          Z.AVL = ****** GX
RE$DAT= 000001          SF.MFL= 000040          T$LLDL  000012          T2.CKD= 010000          Z.DAT = ****** GX
RE$DRP= 000016          SF.PAC= 000020          T$LLDO  000012          T2.DST= 040000          Z.DSP = ****** GX
RE$LDT= 000013          SF.PH3= 020000          T$LLDS  000012          T2.FXD= 002000          Z.FLG = ****** GX
RE$LSN= 000012          SF.PSS= 00004U          T$LLEN  000046          T2.HLT= 000200          Z.NAM = ****** GX
RE$NML= 000001          SF.REA= 000010          T$LOPR  000002          T2.REX= 001000          Z.PCB = ****** GX
RE$OPE= 000004          SF.SEC= 000400          T$LTCL  000024          T2.SEF= 004000          $BIAS = ****** GX
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ADJDF$	#5-60	5-62								
ANBDF$	#5-58	5-63								
ASL$	#5-59									
ASR$	#5-58	10-384	15-792							
CALL	8-241	8-277	8-286	8-299	8-304	8-306	8-307	9-338	9-341	9-345
	10-387	11-434	11-444	11-452	11-457	12-486	12-490	12-493	12-494	13-516
	14-569	14-571	14-590	14-600	14-606	14-621	14-626	14-628	14-629	14-636
	14-643	14-647	14-655	15-710	15-776	15-870	16-926	19-1075	19-1099	19-1118
	19-1120	19-1123	19-1124	19-1139	19-1153	19-1154	19-1155	19-1156	19-1157	19-1158
	19-1163	19-1166	19-1172	19-1193	19-1195	19-1199	19-1201	19-1203	19-1223	19-1237
	19-1253	19-1254	19-1271	19-1272	19-1290	19-1296	19-1297	19-1298	19-1299	19-1301
	19-1308	19-1329	19-1331	20-1369	23-1456	24-1476	24-1481	24-1493	25-1519	26-1539
	26-1544	26-1549	27-1578	27-1581	28-1664	28-1676	28-1694	28-1698	28-1707	28-1718
	28-1723	28-1725	28-1765	28-1767	29-1823	29-1847	29-1855	29-1891	31-1967	31-1975
	31-1980	32-2004	32-2024	32-2031	32-2048					
CALLR	#5-60	19-1256	19-1333							
CCBDF$	#5-60	5-76								
CRASH	8-316	8-317	11-465	11-466	14-663	14-664	15-888	15-889	19-1346	28-1667
	30-1949									
CTRDF$	#5-59	5-64								
DECL$S	#5-59	8-292								
DHBDF$	#5-58	5-73								
DIR$	#8-292	8-292	#8-293	8-293	#8-294	8-294	#9-354	9-354		
ECDDB$	#5-59	5-65								
EVLDF$	#5-59	5-66								
FLTDF$	#5-58	5-67								
FNBDF$	#5-60	5-75								
MOV$	#8-293	8-293	8-293	8-293	8-293	#8-294	8-294			
MRKT$S	#5-58	8-293								
MTPS	13-518	13-543	15-800	15-803	19-1079	19-1087				
NKRDF$	#5-58	5-68								
OBJDF$	#5-58	5-70								
RESMAP	#5-101	24-1495	26-1551	31-1984	32-2051					
RESRG	#5-59	31-1981								
RETURN	8-288	8-311	9-350	9-359	10-388	11-460	12-497	13-545	14-658	15-873
	15-883	16-933	17-1011	18-1051	19-1088	19-1198	19-1210	19-1259	19-1276	19-1339
	20-1377	23-1457	24-1496	25-1520	26-1552	26-1554	27-1649	28-1781	29-1865	29-1892
	30-1945	31-1985	32-2006	32-2052						
RNBDF$	#5-58	5-71								
RVP$	#8-293	8-293								
SAVMAP	#5-97	24-1477	26-1540	31-1968	32-2025					
SAVRG	#5-59	31-1979								
SERDF$	#5-60	5-74								
SLTDF$	#5-59	5-72								
SOB	11-430	18-1045	18-1050	27-1643	28-1757	28-1775				
STMDF$	#5-60	5-77								
SWSTK$	8-277	9-338	13-516	15-710	19-1075	19-1193	19-1201	19-1253	19-1271	19-1329
	20-1369	24-1476	26-1539	27-1578	28-1723	28-1765	29-1823	31-1967	32-2024	
TCBDF$	#5-80	5-81								
WSIG$S	#5-60	9-354								
WTSE$S	#5-58	8-294								
XPDDB$	#5-59	5-69								

```
                        .TITLE  POLAL - NETWORK POOL ALLOCATION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
; NTL - ROUTINES FOR SINGLE WORD ALLOCATIONS FROM THE NETWORK POOL
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

POLAL CREATED BY MACRO ON 29-JUN-85 AT 00:36 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES								
CALL	7-131	7-154	7-168	8-215	8-238	8-250	8-267	9-328	9-333
MAP	#5-61	#5-65	8-258	8-271	8-281	8-283	8-287		
RESMAP	#5-73	7-172	8-260	8-273	8-294				
RESRG	#5-61	8-295	10-397						
RETURN	7-174	8-296	9-339	10-401					
SAVMAP	#5-69	7-119	8-233	8-240	8-257	8-265	8-280		
SAVRG	#5-61	8-203	10-365						

FILEID**PRALL

```
                        .TITLE  PRIO - READ PROCESS INTO CORE
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - COMMON "SET CEX" AND "SET PROCESS" ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

```
570 001320  000746                          BR      50$             ;*
571                                         .IF DF  M$$MGE
572 001322  012767  000034  000010' 151$:   MOV     #ERR28,ERROR    ;* ILLEGAL COMM EXEC PARTITION
573 001330  000742                          BR      50$             ;*
574                                         .ENDC
```

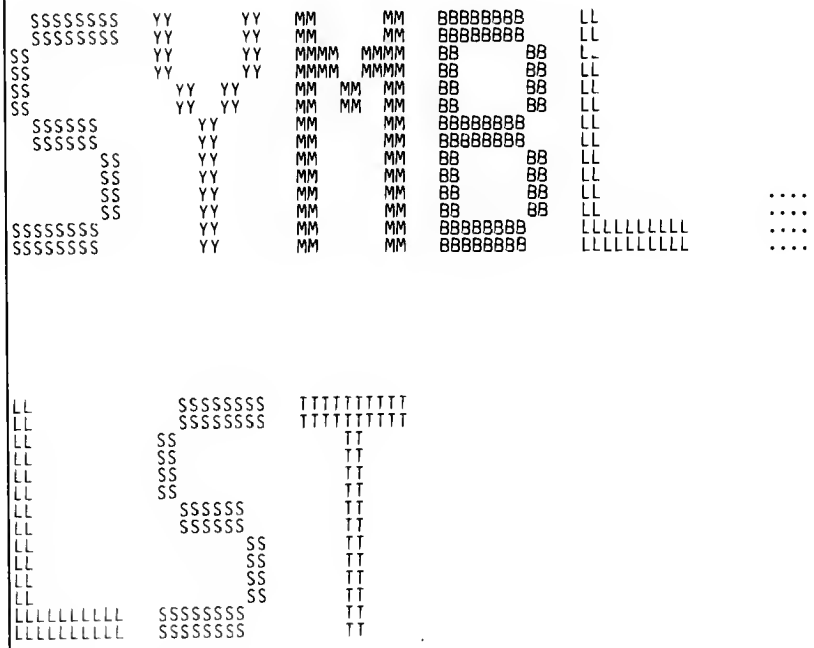
Table of contents

5- 55 LOCAL DATA
6- 71 $ALOC2 - ALLOCATE POOL SPACE
7- 127 $DEAC2 - DEALLOCATE FROM END OF PARTITION

```
                                        .SBTTL  LOCAL DATA
                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                        .NLIST  BEX


                                ; "NTLCEX"/"NTLPRO" ERROR MESSAGES
                                ;
                                        .ENABL  LC

000000                                  NTLER$  <:>,44,4,PERR,$PRV.N,$FNOUT,<Open Failure (-***.)>
000036                                  NTLER$  <:>,45,4,PERR,$PRV.N,$FNOUT,<Read Failure (-***.)>
000074                                  NTLER$  <:>,46,4,PERR,$REP.P,$FNOUT,<Read Overflow>
000122                                  NTLER$  <:>,47,4,PERR,$REP.P,$FNOUT,<Format Error>
000150                                  NTLER$  <:>,48,,PERR,RTSPC,,<Comm Exec Incompatible With RSX-11M System>
000234                                  NTLER$  <:>,49,4A,PERR,REP4A,,<Undefined>
000256                                  NTLER$  <:>,50,5,PERR,$REP.P,,<Incompatible With Comm Exec>
000322                                  NTLER$  <:>,51,5,PERR,$PRV.N,$FNOUT,<Open Failure (-***.)>
000360                                  NTLER$  <:>,53,5,PERR,$REP.P,$FNOUT,<Not Contiguous>
000410                                  NTLER$  <:>,54,5,PERR,$PRV.N,$FNOUT,<Label Block Read Failure (-***.)>
000462                                  NTLER$  <:>,55,5,PERR,$REP.P,$FNOUT,<Not a Valid Binary Image>
                                ;       .IF DF  M$$MGE
000524                          ;       NTLER$  <:>,56,5,PERR,$REP.P,$FNOUT,<Not Built For APR5>
000560                                  NTLER$  <:>,57,5,PERR,$REP.P,$FNOUT,<Size Exceeds 4K Words>
000616                                  NTLER$  <:>,58,5,PERR,$REP.P,$FNOUT,<Extension Exceeds 4K Words>
                                ;       .ENDC
000662                          ;       NTLER$  <:>,59,5A,PERR,REP5A,,<Not In System>
000710                                  NTLER$  <:>,60,5A,PERR,REP5A,,<Base Address Mismatch>
000746                                  NTLER$  <:>,61,5A,PERR,REP5A,,<Too Small>
000770                                  NTLER$  <:>,62,5A,PERR,REP5A,,<Not Common>
001014                                  NTLER$  <:>,63,5A,PERR,REP5A,,<Busy>
                                ;       .IF DF  M$$MGE
001032                          ;       NTLER$  <:>,64,5A,PERR,REP5A,,<Not In Exec Space>
001064                                  NTLER$  <:>,65,5A,PERR,REP5A,,<Sub-PCB Allocation Failure>
001130                                  NTLER$  <:>,66,5A,PERR,REP5A,,<Too Fragmented>
                                ;       .ENDC
001160                          ;       NTLER$  <:>,67,5,PERR,$PRV.N,$FNOUT,<Read Failure (-***.)>
001216                                  NTLER$  <:>,68,,PERR,RTSPC,,<CETAB Allocation Failure>
001260                                  NTLER$  <:>,69,,PERR,RTSPC,,<Minhop/mincost allocation failure>
001332                                  NTLER$  <:>,70,,PERR,RTSPC,,<Node counter allocation failure>
001402                                  NTLER$  <:>,ZA,,PERR,RTSPC,,<Incompatible With Comm Exec>
001446                                  NTLER$  <:>,XQ,,PERR,RTSPC,,<CCB Allocation Failure>
001506                                  NTLER$  <:>,QX,,PERR,RTSPC,,<Timer cells not found>
001544                                  NTLER$  <:>,QY,,PERR,RTSPC,,<System not built with network support>
                                        .IF DF  R$$11D!I$$AS
                                        NTLER$  <:>,QU,0,PERR,RTSPC,,<Failed to find Comm/Exec Region>
                                        NTLER$  <:>,QV,5C,PERR,REP5A,,<Creation Failed>
                                        NTLER$  <:>,QW,5C,PERR,REP5B,,<Mapping Failed>
                                        NTLER$  <:>,QZ,,PERR,RTSPC,,<Unable to run task -- COMDRV>
                                        .ENDC
                                        .IF DF  R$$MPL
                                        NTLER$  <:>,9A,,PERR,RTSPC,,<Reconfiguration device not in system>
                                        NTLER$  <:>,9Z,9,PERR,RTSPC,,<Failed to re-initialize HRC>
```

```
                                        .SBTTL  MACRO DEFINITIONS

                                        .MCALL  XPDDB$,CTRDF$,BIAS,ECDDB$,DHBDF$,SAVRG,RESRG,CACDF$,ADJDF$
                                        .MCALL  EMSG$,ASR$

                                        .MACRO  SAVMAP
                                        MOV     @KSAR5,-(SP)
                                        .ENDM

                                        .MACRO  RESMAP
                                        MOV     (SP)+,@KSAR5
                                        .ENDM

                                        .MACRO  BIAS5 REG
                                        BIC     #160000,REG
                                        BIS     #120000,REG
                                        .ENDM

                                        .MACRO  MAP BIAS
                                        MOV     BIAS,@KSAR5
                                        .ENDM


000000                                  ADJDF$                  ; DEFINE ADJACENCY DATA BASE SYMBOLS
000000                                  CTRDF$                  ; DEFINE COUNTER BLOCK SYMBOLS
000000                                  ECDDB$                  ; DEFINE ECL DATA BASE SYMBOLS
000000                                  XPDDB$                  ; DEFINE XPT DATA BASE SYMBOLS
000000                                  DHBDF$                  ; DEFINE DECNET HOME BLOCK SYMBOLS
000000                                  CACDF$                  ; DEFINE CACHE SYMBOLS


000000                                  .PSECT
```

```
**FILE**ID**SYMBL
```



```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  EMSG$,EMSG$R,NTLER$,SLTDF$

000000                                  SLTDF$                  ; DEFINE SLT OFFSETS AND SYMBOLS

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D


                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000000                                  NTLER$  ,2,2,$TERR,$REP.C,$ELINE,<Only One Operand Allowed>
000042                                  NTLER$  ,14,2,$TERR,$REP.C,$ELINE,<Length Verification Failure>
000106                                  NTLER$  ,15,2,$TERR,$REP.C,$ELINE,<Offset Value Too Large>
000146                                  NTLER$  ,18,2,$TWARN,$REP.C,$ELINE,<Expression Value Truncation>
000212                                  NTLER$  ,19,2,$TWARN,$REP.C,$ELINE,<Modification Value Truncation>
000260                                  NTLER$  ,20,2,$TERR,$REP.C,$ELINE,<File Name Too Long>
000314                                  NTLER$  ,52,2,$TERR,$REP.C,$ELINE,<File Name May Not Contain a Period>
000370                                  NTLER$  ,80,2,$TERR,$REP.C,$ELINE,<DDM/DLC Operator>

                                        .DSABL  LC


                                ; ERROR MESSAGE FORMAT STRING
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000422  052     040     124     FMT2:   .ASCIZ  "* Template -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

000000                                  .PSECT
```

```
                                        ;+
                                        ; $ADDR8 - ".ADDRB" ACTION ROUTINE
                                        ; $SLN8 - ".SLNB" ACTION ROUTINE
                                        ; $STN8 - ".STNB" ACTION ROUTINE
                                        ; $UNB - ".UNB" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2=DESTROYED
                                        ;-
                                        ;
                                                .ENABL  LSB
                                        ;
000604                                  $ADDRB::
000604  012701  000000G                         MOV     #.ADDRB,R1      ; SET FUNCTION CODE
000610  000410                                 BR      10$
                                        ;
000612                                  $STNB::
000612  012701  000000G                         MOV     #.STNB,R1       ; SET FUNCTION CODE
000616  000405                                 BR      10$
000620                                  $SLNB::
000620  012701  000000G                         MOV     #.SLNB,R1       ; SET FUNCTION CODE
000624  000402                                 BR      10$
000626                                  $UNB::
000626  012701  000000G                         MOV     #.UNB,R1        ; SET FUNCTION CODE
000632                                  10$:    CALL    $VXLLC          ; MAKE SURE IT'S NOT AN LLC
000636                                  15$:    CALL    $RQ1            ; REQUEST BINARY FILE
000642  103403                                 BCS     20$             ; IF CS, NO ACCESS
000644  110112                                 MOVB    R1,(R2)         ; STORE FUNCTION CODE
000646  005267  000000G                         INC     $SIZE           ; ADJUST DATA BASE SIZE
000652                                  20$:    RETURN
                                        ;
                                                .DSABL  LSB
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
L.NMST	000020		#5-61								
L.NSTA	000014		#5-61								
L.OWNR	000021		#5-61								
L.UNT	000013		#5-61								
MAXOV	= ******	GX	22-536								
SF.ACT	= 000200		#5-61								
SF.ENA	= 000100		#5-61								
SF.LPB	= 000004		#5-61								
SF.MFL	= 000040		#5-61								
SF.PAC	= 000020		#5-61								
SF.REA	= 000010		#5-61								
SF.SER	= 000001		#5-61								
SF.SVC	= 000002		#5-61								
SF.UNL	= 000040		#5-61								
S$$BAS	= ******		5-74	5-74	5-75	5-75	5-76	5-76	5-77	5-77	5-78
			5-78	5-79	5-79	5-80	5-80	5-81	5-81		
S.COST	000001		#5-61								
S.FLG	000000		#5-61								
S.LEN	000004		#5-61								
S.NMST	000002		#5-61								
S.OWNR	000003		#5-61								
$ADDRB	000604	RG	#18-381								
$ADDRW	000654	RG	#19-418								
$APR	000734	RG	#20-453								
$BLKB	000000	RG	#6-110								
$BLKW	000014	RG	#6-115								
$BYTE	000102	RG	#7-150	8-172							
$CNB	000126	RG	#8-168								
$CNW	000150	RG	#8-174								
$CSR	000176	RG	#9-191								
$CTM	000662	RG	#19-422								
$DIFF	= ******	GX	9-196	*24-593	24-596	*24-600	*25-617				
$DPRB	000742	RG	#20-456								
$DPRW	000770	RG	#20-479								
$DVCHA	000230	RG	#10-209								
$ELINE	= ******	GX	5-74	5-75	5-76	5-77	5-78	5-79	5-80	5-81	
$END1	000304	RG	#12-244								
$END2	000340	RG	#13-263								
$ERR14	000042	R	#5-75	12-251							
$ERR15	000106	R	#5-76	13-278							
$ERR18	000146	R	#5-77	23-576							
$ERR19	000212	R	#5-78	24-603							
$ERR2	000000	R	#5-74	13-277							
$ERR20	000260	R	#5-79	26-649							
$ERR52	000314	R	#5-80	26-650							
$ERR80	000370	R	#5-81	27-664							
$FILEN	= ******	GX	26-634								
$FLAGS	= ******	GX	13-266	27-662							
$FLAG1	= ******	GX	12-245	13-268							
$FLAG2	= ******	GX	23-564	23-568	23-572	*23-577	24-598	*24-604	*26-643		
$FNAME	001350	RG	#26-632								
$LFLD	= ******	GX	*14-295								

```
                                        .SBTTL  DQCLK - DEQUEUE 100MSEC CLOCK BLOCK
                                ;+
                                ; DQCLK - DEQUEUE 100MSEC CLOCK BLOCK
                                ;
                                ; THIS ROUTINE WILL DEQUEUE A 100MSEC CLOCK BLOCK FOR THIS ZF.INI PROCESS
                                ;
                                ; INPUTS:
                                ;       R4 - PDV ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       DEQUEUE 100MSEC CLOCK BLOCK IF ONE EXISTS
                                ;
                                ;-
000766                          DQCLK:  SWSTK$  30$             ; ENTER SYSTEM STATE
000772  016700  000000G               MOV     T100Q,R0        ; GET SHORT TIMER LISTHEAD
000776                                MTPS    #PR7            ; DISABLE INTERRUPTS
001004  011003                10$:    MOV     (R0),R3         ;;; GET FIRST BLOCK
001006  001432                        BEQ     20$             ;;; IF END - BRANCH
001010  010005                        MOV     R0,R5           ;;; COPY PREVIOUS BLOCK ADDRESS
001012  010300                        MOV     R3,R0           ;;; COPY NEXT BLOCK ADDRESS
001014  017701  000000G               MOV     @PDVTA,R1       ;;; GET $PDVTA CONTENTS
001020  005002                        CLR     R2              ;;; INIT REG
001022  156302  000002                BISB    X.ID(R3),R2     ;;; GET PDV INDEX
001026  060201                        ADD     R2,R1           ;;; GET PDV ADDRESS
001030  020411                        CMP     R4,(R1)         ;;; IS IT PROCESS BEING UNLOADED
001032  001364                        BNE     10$             ;;; IF NO - BRANCH
001034  011315                        MOV     (R3),(R5)       ;;; UNLINK CLOCK BLOCK
001036  001016                        BNE     20$             ;;; IF NOT LAST BLOCK - BRANCH
001040  016701  000000G               MOV     T100Q,R1        ;;; GET HEAD OF LIST
001044  062701  000002                ADD     #2,R1           ;;; GET ADDRESS OF $T100Q+2
001050  010511                        MOV     R5,(R1)         ;;; UPDATE NEW LAST BLOCK IN LIST
001052  016701  000000G               MOV     CLKHD,R1        ;;; GET START OF CLOCK QUEUE
001056  010102                15$:    MOV     R1,R2           ;;; COPY PREVIOUS BLOCK ADDRESS
001060  011101                        MOV     (R1),R1         ;;; GET NEXT BLOCK
001062  001404                        BEQ     20$             ;;; IF END OF LIST - BRANCH

001064  026701  000000G               CMP     T100C,R1        ;;; IS IT OUR TIMER BLOCK ?
001070  001372                        BNE     15$             ;;; IF NO GET NEXT BLOCK
001072  011112                        MOV     (R1),(R2)       ;;; UNLINK OUR TIMER BLOCK

001074                          20$:    MTPS    #0              ;;; ENABLE INTERRUPTS

001100                          30$:    RETURN
```

```
1110 002302  005764  000000G                TST     Z.PCB(R4)          ; IS PROCESS CORE-RESIDENT?
1111 002306  001426                         BEQ     50$                ; IF EQ, NO
1112
1113 002310  032764  000000G 000000G        BIT     #ZF.INI,Z.FLG(R4) ; DOES PROCESS NEED TERMINATION CALL ?
1114 002316  001416                         BEQ     51$                ; IF NO - BRANCH
1115 002320  012767  003030  000000G        MOV     #FS.TRM!FC.PCT,$$PDIN ; STORE FUNCTION CODE
1116 002326  010401                         MOV     R4,R1              ; COPY PDV ADDRESS
1117 002330  016467  000000G 000000G        MOV     Z.PCB(R4),$PCB     ; SAVE PCB A ADDRESS
1118 002336                                 CALL    DQCLK              ; STOP 100MSEC TIMER
1119
1120 002342                                 CALL    $DSINI             ; DISPATCH TO PROCESS
1121 002346  005064  000000G                CLR     Z.PCB(R4)          ; STOP ONE SEC TIMER
1122 002352  000241                         CLC                        ; CLEAR C-BIT
1123 002354                         51$:    CALL    XDCB               ; POSSIBLE UNLOAD OF RSX11M DEVICE DRIVER
1124 002360                                 CALL    XPRO               ; RELEASE PROCESS FROM MEMORY
1125 002364  005316                 50$:    DEC     (SP)               ; LOOP <$PDVNM> TIMES
1126 002366  001343                         BNE     40$                ;   ...
1127 002370  017700  000000G        52$:    MOV     @CCBLH,R0          ; GET FREE CCB ADDRESS
1128 002374  001407                         BEQ     54$                ; IF ZERO, NO MORE CCB'S
1129 002376  011077  000000G                MOV     (R0),@CCBLH        ; SET NEW FIRST IN LIST
1130 002402  017701  000000G                MOV     @CCBSZ,R1          ; GET CCB SIZE
1131
1132                                        .IF DF  R$$MPL
1133                                        TSTB    @NCPU              ; MULTIPROCESSOR SYSTEM?
1134                                        BEQ     53$                ; BR IF NO
1135                                        TST     -(R0)              ; DE-ALLOCATE 2 EXTRA BYTES (NEGATIVE OFFSET)
1136                                        ADD     #2,R1              ; ADD 2 TO NUMBER OF BYTES TO DE-ALLOCATE
1137                                53$:    .ENDC
1138
1139 002406                                 CALL    $DEA16             ; DE-ALLOCATE A CCB
1140 002412  000766                         BR      52$                ; KEEP LOOPING
1141 002414  016701  000000G        54$:    MOV     RDBLH,R1           ; GET $RDBLH ADDRESS
1142 002420  005021                         CLR     (R1)+              ; CLEAR OUT LISTHEADS
1143 002422  005011                         CLR     (R1)               ;   FOR BUFFER ALLOCATIONS
1144 002424  016701  000000G                MOV     SDBLH,R1           ; GET $SDBLH ADDRESS
1145 002430  005021                         CLR     (R1)+              ; INIT LISTHEADS
1146 002432  005011                         CLR     (R1)               ;   ...
1147 002434  005726                         TST     (SP)+              ; PURGE STACK
1148
1149                                        ;
1150                                        ; CLEAN-UP THE POOL TABLES (NOT TO BE CONFUSED WITH BAR BILLIARDS TABLES)
1151                                        ;
1152
1153 002436                                 CALL    XOBJTB             ; DE-ALLOCATE ALL OBJECTS
1154 002442                                 CALL    XALITB             ; DE-ALLOCATE ALL ALIASES
1155 002446                                 CALL    XREMTB             ; DE-ALLOCATE ALL REMOTES
1156 002452                                 CALL    XFLTTB             ; DE-ALLOCATE ALL FILTER CONTROL BLOCKS
1157 002456                                 CALL    XSERTB             ; DE-ALLOCATE SERVICE BLOCK
1158 002462                                 CALL    XFILTB             ; DE-ALLOCATE FILENAME BLOCK
1159
1160
1161 002466  005727  000000G                TST     #$XPSI             ; PSI SUPPORTED?
1162 002472  001402                         BEQ     55$                ; NO
1163 002474                                 CALL    $XPSI              ; DEALLOCATE PSI DATABASES
1164 002500  005727  000000G        55$:    TST     #$XSNA             ; SNA SUPPORTED?
1165 002504  001402                         BEQ     56$                ; NO
1166 002506                                 CALL    $XSNA              ; DEALLOCATE SNA DATABASES
```

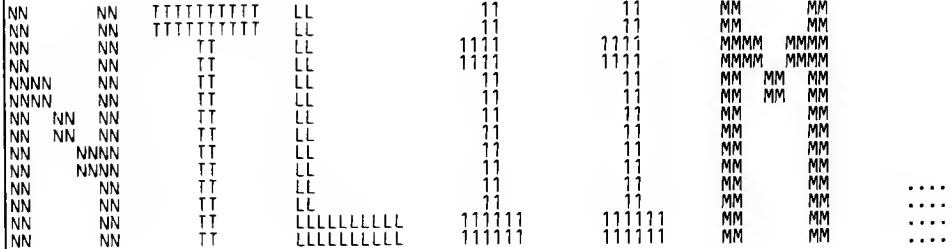
```
1613 003470  032702  000077          50$:    BIT     #77,R2          ;; IS IT PHYSICAL END OF BLOCK?
1614 003474  001403                          BEQ     60$             ;; IF EQ, YES
1615 003476  005712                          TST     (R2)            ;; IS IT LOGICAL END OF BLOCK?
1616 003500  001340                          BNE     20$             ;; IF NE, NO - CHECK NEXT ENTRY
1617 003502  000402                          BR      70$             ;;
1618 003504  162702  000100          60$:    SUB     #100,R2         ;; RESET R2 TO START OF BLOCK
1619 003510  042702  000077          70$:    BIC     #77,R2          ;;      ...
1620
1621                                         .IF DF  M$$MGE
1622 003514  011211                          MOV     (R2),(R1)       ;; MAP TO THE NEXT UNLOAD BLOCK
1623                                         .IFF
1624                                         MOV     (R2),R2         ; GET NEXT UNLOAD BLOCK ADDRESS
1625                                         .ENDC
1626
1627 003516  001330                          BNE     10$             ;; IF NE, THERE IS ANOTHER ONE
1628 003520  000416                          BR      100$            ;;
1629 003522  024242                  80$:    CMP     -(R2),-(R2)     ;; RESET R2 TO FIRST WORD OF ENTRY
1630 003524  010267  000012'                 MOV     R2,UBA          ;; SAVE UNLOAD ENTRY ADDRESS
1631
1632                                         .IF DF  M$$MGE
1633 003530  011167  000010'                 MOV     (R1),UBIAS      ;; AND BIAS
1634                                         .IFTF
1635 003534  042702  000077                  BIC     #77,R2          ;; RESET R2 TO START OF BLOCK
1636                                         .IFF
1637                                         MOV     R2,UBIAS        ; SAVE START OF BLOCK ADDRESS
1638                                         .ENDC
1639
1640 003540  012700  000014'                 MOV     #BLOCK,R0       ;; MAKE A LOCAL COPY OF THE UNLOAD BLOCK
1641 003544  012701  000040                  MOV     #32.,R1         ;;      ...
1642 003550  012220                  90$:    MOV     (R2)+,(R0)+     ;;      ...
1643 003552                                  SOB     R1,90$          ;;      ...
1644
1645 003556                          100$:   .IF DF R$$11D ! I$$AS
1646                                         CALL    @(SP)+          ;; RESTORE PREVIOUS MAPPING
1647                                         .ENDC
1648
1649 003556                                  RETURN                  ;; BACK TO USER MODE AND THEN TO CALLER
```

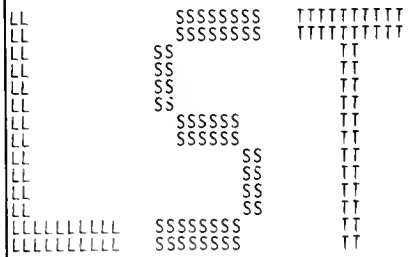
NTLUNL - NTL UNLOAD ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:34 Page 32-5
Symbol table

```
$CCEX   002134RG        $DLCPV= ****** GX       $PRAVL= ****** GX       $XPSI  = ****** GX       .CXUNL= ****** GX
$CEACX= ****** GX       $DSINI  000336RG        $SAVRG= ****** GX       $XSNA  = ****** GX       .DEA16  004166R
$CLIN   001102RG        $FLAGS= ****** GX       $SLTA = ****** GX       $$PDIN= ****** GX       .NAME   000146R     002
$CPRO   000674RG        $ICBLN= ****** GX       $XDEAC= ****** GX       $$$    = 000062         .NTLPT  000150R     002
$DDMPV= ****** GX       $MAPX = ****** GX       $XLBR = ****** GX       $$$ARG= 000004          .PCB   = ****** GX
$DEAHC= ****** GX       $MXVCT= ****** GX       $XLIN   000476RG        .CXALL= ****** GX       .UMRBA= ****** GX
$DEA16= ****** GX       $NAME = ****** GX       $XPAR = ****** GX       .CXLEN= ****** GX       .$$$$.= 000034
$DEA18= ****** GX       $PCB  = ****** GX       $XPRO   000000RG        .CXPCB= ****** GX       ..PDV   000152R     002
$DECEX= ****** GX       $PDVA = ****** GX

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        004626    001   (RW,I,LCL,REL,CON)
DATA    000154    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 172
Work  file writes: 140
Size of work file: 29607 Words  ( 116 Pages)
Size of core pool: 17608 Words  ( 67 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:01:07.17
SY:NTLUNL.V2,[132,134]NTLUNL/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]NTLUNL
```


NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 1

FILEID**NTL11M


NN NN TTTTTTTTTT LL 11 11 MM MM
NN NN TTTTTTTTTT LL 11 11 MM MM
NN NN TT LL 1111 1111 MMMM MMMM
NN NN TT LL 1111 1111 MMMM MMMM
NNNN NN TT LL 11 11 MM MM MM
NNNN NN TT LL 11 11 MM MM MM
NN NN NN TT LL 11 11 MM MM
NN NN NN TT LL 11 11 MM MM
NN NNNN TT LL 11 11 MM MM
NN NNNN TT LL 11 11 MM MM
NN NN TT LL 11 11 MM MM ....
NN NN TT LL 11 11 MM MM ....
NN NN TT LLLLLLLLLL 111111 111111 MM MM ....
NN NN TT LLLLLLLLLL 111111 111111 MM MM ....



```
                        .SBTTL  MACROS


;
; GLOBAL MACROS
;

        .MCALL  MAP,SAVRG,RESRG

;
; LOCAL MACRO DEFINTIONS
;
        .MACRO  MAP BIAS
        MOV     BIAS,@KSAR5     ; SET UP APR5 MAPPING
        .ENDM

        .MACRO  SAVMAP
        MOV     @KSAR5,-(SP)    ; SAVE APR 5
        .ENDM

        .MACRO  RESMAP
        MOV     (SP)+,@KSAR5    ; RESTORE APR5
        .ENDM
```

FILEID**PRALL

```
PPPPPPPP      RRRRRRRR        AAAAAA      LL            LL
PPPPPPPP      RRRRRRRR        AAAAAA      LL            LL
PP       PP   RR       RR    AA      AA   LL            LL
PP       PP   RR       RR    AA      AA   LL            LL
PP       PP   RR       RR    AA      AA   LL            LL
PP       PP   RR       RR    AA      AA   LL            LL
PPPPPPPP      RRRRRRRR       AA      AA   LL            LL
PPPPPPPP      RRRRRRRR       AA      AA   LL            LL
PP            RR  RR         AAAAAAAAAA   LL            LL
PP            RR  RR         AAAAAAAAAA   LL            LL
PP            RR    RR       AA      AA   LL            LL                ....
PP            RR    RR       AA      AA   LL            LL                ....
PP            RR      RR     AA      AA   LLLLLLLLLL    LLLLLLLLLL        ....
PP            RR      RR     AA      AA   LLLLLLLLLL    LLLLLLLLLL        ....


LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

```
                                        .SBTTL   MACRO DEFINITIONS


                                ;
                                ; LIBRARY MACROS
                                ;
                                        .MCALL  NHWDF$,RESRG,SAVRG,CACDF$,RETC,ECDDB$

                                        .MCALL  LBLDF$,PCBDF$,DHBDF$,CALLR

000000                                  LBLDF$                  ; DEFINE TASK IMAGE LABEL BLOCK OFFSETS
000000                                  PCBDF$                  ; PARTITION CONTROL BLOCK OFFSETS
000000                                  DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS
000000                                  CACDF$                  ; DEFINE NI CACHE OFFSETS
000000                                  ECDDB$                  ; DEFINE ECL DDB OFFSETS


                                ;
                                ; LIBRARY SYMBOLS
                                ;
000000                                  NHWDF$                  ; FEATURES MASK BIT DEFINITIONS

                                ;
                                ; LOCAL MACROS
                                ;


                                ;
                                ; SHIFT LEFT
                                ;
                                        .MACRO  ASL$    COUNT,REG
                                        .IF DF  R$$EIS
                                        ASH     #'COUNT',REG
                                        .IFF
                                        .REPT   COUNT
                                        ASL     REG
                                        .ENDR
                                        .ENDC
                                        .ENDM   ASL$

                                        .MACRO  ASR$    COUNT,REG
                                        .IF DF  R$$EIS
                                        ASH     #-'COUNT',REG
                                        .IFF
                                        .REPT   COUNT
                                        ASR     REG
                                        .ENDR
                                        .ENDC
                                        .ENDM   ASR$
```

```
                                        .SBTTL  PR.SUB - CREATE SUB-PARTITION

                                ;+
                                ;
                                ; PR.SUB - CREATE SUB-PARTITION
                                ;
                                ; THIS ROUTINE IS CALLED IF THE PROCESS PARTITION IS SYSTEM CONTROLLED.
                                ; A NEW SUB-PARTITION IS CREATED AND LINKED IN. THE PRESENCE OF AN MCB
                                ; PROCESS IN A SYSTEM CONTROLLED PARTITION IS DENOTED BY SETTING THE COMMON
                                ; BIT (PS.COM) IN THE PCB STATUS WORD (P.STAT) AND BY SETTING THE PARTITION
                                ; NAME (P.NAM) TO 'NT.***', WHERE '***' IS THE THREE CHARACTER RAD50 PDV NAME.
                                ;
                                ; INPUTS:
                                ;       R5=MAIN PARTITION PCB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2,R3,R4=DESTROYED
                                ;       R5=SUB-PARTITION PCB ADDRESS
                                ;-
                                        .IF DF M$$MGE

001332  016701  000000G        PR.SUB: MOV     PLGTH,R1        ;* GET PCB LENGTH
001336                                  CALL    @ALOCB          ;* TRY TO ALLOCATE SUB-PARTITION PCB
001342  103512                          BCS     101$            ;* IF CS, FAILURE
001344  010001                          MOV     R0,R1           ;* COPY PCB ADDRESS
001346  016702  000000G                 MOV     PLGTH,R2        ;* GET PCB LENGTH
001352  006202                          ASR     R2              ;* CONVERT TO A WORD COUNT
001354  005021                  5$:     CLR     (R1)+           ;* ZERO THE ENTIRE SUB-PCB
001356                                  SOB     R2,5$           ;*  ...
001362  012760  055274  000004        MOV     #^RNT.,P.NAM(R0) ;* SET NAME TO 'NT.***'
001370  016760  000000G 000006        MOV     $NAME,P.NAM+2(R0) ;*  ...
001376  010560  000012                MOV     R5,P.MAIN(R0)   ;* MAIN PCB ADDRESS
001402  016760  000[illegible] 000016  MOV     BLOX,P.BLKS(R0) ;* SIZE IN 32. WORD BLOCKS

                                        .IF DF  R$$MPL

                                        MOV     #PS.COM!PS.NSF!PS.FXD!PS.CHK,P.STAT(R0) ;* COMMON, NO SHUFFLING
                                                                                        ;* FIXED, NO CHECKPOINTING
                                        MOV     #P2.RON,P.ST2(R0) ;* MAKE PARTITION READ-ONLY

                                        .IFF    ; DF R$$MPL

001410  012760  000640  000030        MOV     #PS.COM!PS.NSF!PS.SYS,P.STAT(R0) ;* COMMON, NO SHUFFLING

                                        .ENDC

001416  010501                          MOV     R5,R1           ;* COPY MAIN PCB ADDRESS
001420  005002                          CLR     R2              ;* NO PREVIOUS SUB-PCB
001422  005046                          CLR     -(SP)           ;* PREVIOUS SUB-PCB IF SPACE IS FOUND
001424  066102  000014          10$:    ADD     P.REL(R1),R2    ;* GIVING BIAS OF NEXT HOLE IN PARTITION
001430  016103  000010                  MOV     P.SUB(R1),R3    ;* GET NEXT SUB-PCB
001434  001425                          BEQ     30$             ;* IF ZERO, END OF LIST
001436  016304  000014                  MOV     P.REL(R3),R4    ;* GET BIAS OF SUB-PARTITION
001442  160204                          SUB     R2,R4           ;* GIVING SIZE OF HOLE
001444  166704  000000'                 SUB     BLOX,R4         ;* IS THE HOLE BIG ENOUGH?
001450  103413                          BLO     20$             ;* IF MI, NO
```

```
                        .TITLE  PROAL - PROCESS ALLOCATION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       EXTERNAL MAPPING ALLOCATION/DEALLOCATAION ROUTINES
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

LOCAL DATA

```
                                        .ENDC
001622                                  NTLER$  <:>,99,,PERR,RTSPC,,<Reachability table allocation failure>
001700                                  NTLER$  <:>,9C,,PERR,RTSPC,,<ECL not found>
001726                                  NTLER$  <:>,9D,,PERR,RTSPC,,<Network POOL.. block allocation failure>
002006                                  NTLER$  <:>,9E,,PERR,RTSPC,,<Timer service initialization failure>

                                        .DSABL  LC


                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
002064  000                     FMT0:   .ASCIZ  ""
002065  052     040     123     FMT4:   .ASCIZ  "* Symbol Table File -- "
002115  052     040     123     FMT4A:  .ASCIZ  "* Symbol Table File -- Symbol * "
002156  052     040     120     FMT5:   .ASCIZ  "* Process File -- "
002201  052     040     120     FMT5A:  .ASCIZ  "* Process File -- Partition * "
002240  052     040     000     FMT5B:  .ASCIZ  "* "
                                        .IF DF  R$$11D!I$$AS
                                FMT5C:  .ASCIZ  "* Process File -- Region *, "
                                        .ENDC
002243  052     127     101     FMT9:   .ASCIZ  "*WARNING* -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX


                                ;
                                ; ERROR MESSAGE BUFFER
                                ;
002262  015                     ERRMSG: .BYTE   15
002263  116     124     114             .ASCII  "NTL -- "
002266  040     055     055
002271  040
002274                          ERRBUF: .BLKB   70.

                                        .EVEN


000000                                  .PSECT
```

```
                                        .SBTTL  $NICAH - ALLOCATE XPT CACHE
                                ;+
                                ; $NICAH - ALLOCATE XPT CACHE
                                ;
                                ; INPUTS:
                                ;       $PDVA - XPR PDV ADDRESS
                                ;       R2    - XPT DDB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       XPT CACHE ALLOCATED
                                ;-

000000                          $NICAH::CALL    $SAVAL                  ;SAVE REGS
000004  017703  000000G                 MOV     @DECPT,R3               ;GET DECNET HOME BLOCK ADDRESS
000010  016300  000060                  MOV     D$NN(R3),R0             ;GET NUMBER OF NODES
000014  001061                          BNE     40$                     ;IF ROUTING NODE - BRANCH
000016  016705  000000G                 MOV     $PDVA,R5                ;STORE PDV ADDRESS
000022  010204                          MOV     R2,R4                   ;COPY XPT DDB ADDRESS
000024                                  SWSTK$  50$

000030  017702  000000G                 MOV     @DECPT,R2               ;;GET DECNET HOME BLOCK ADDRESS
000034  005000                          CLR     R0                      ;;INIT REG
000036  156200  000040                  BISB    D$MLL(R2),R0            ;;GET MAX LINKS
000042  010003                          MOV     R0,R3                   ;;SAVE MAX LINKS
000044  012701  000012                  MOV     #C$LEN,R1               ;;GET CACHE ENTRY SIZE
000050                                  CALL    $MUL                    ;;CALCULATE TOTAL BLOCK SIZE
000054  010100                          MOV     R1,R0                   ;;SAVE RESULT
000056  010364  000062                  MOV     R3,N$CACH(R4)           ;;STORE NUMBER NI CACHE ENTRIES

000062                                  SAVMAP                          ;;SAVE MAPPING
000066  016401  000064                  MOV     N$CACH+2(R4),R1         ;;GET ADDRESS (R0 CONTAINS SIZE)

000072                                  MAP     Z.DSP(R5)               ;;MAP TO XPT
000100  022701  140000                  CMP     #140000,R1              ;;NEXT 4KW BLOCK ?
000104  101003                          BHI     10$                     ;;IF NO - BRANCH

000106  062777  000200  000000G         ADD     #200,@KSAR5             ;;MAP TO NEXT BLOCK
000114                          10$:    BIAS5   R1                      ;;BIAS APR5

000124  105021                  20$:    CLRB    (R1)+                   ;;CLEAR XPT CACHE
000126  022701  140000                  CMP     #140000,R1              ;;NEXT BLOCK ?
000132  001005                          BNE     30$                     ;;IF NO - BRANCH

000134  062777  000200  000000G         ADD     #200,@KSAR5             ;;MAP TO NEXT BLOCK
000142  012701  120000                  MOV     #120000,R1              ;;BIAS APR5

000146                          30$:    SOB     R0,20$                  ;;IF MORE - BRANCH

000152                                  RESMAP                          ;;RESTORE MAPPING
000156                                  RETURN

000160  005062  000062          40$:    CLR     N$CACH(R2)              ;CLEAR CACHE ENTRIES
000164  005062  000064                  CLR     N$CACH+2(R2)            ;CLEAR CACHE ADDRESS
```

```
                        .TITLE  SYMBL - SYMBOL TABLE FILE LOOKUP ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       SYMBOL TABLE FILE LOOKUP ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                                        ;+
                                        ; $BLKB - ".BLKB" ACTION ROUTINE
                                        ; $BLKW - ".BLKW" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=NUMBER OF BYTES/WORDS
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;-
                                                .ENABL  LSB

000000                                  $BLKB::
000000  012700  000000G                         MOV     #.BLKB,R0       ; SET FUNCTION CODE
000004  016701  000000G                         MOV     $VALUE,R1       ; GET BYTE COUNT
000010  001431                                 BEQ     50$             ; IF ZERO, DO NOTHING
000012  000407                                 BR      10$
000014                                  $BLKW::
000014  012700  000000G                         MOV     #.BLKW,R0       ; SET FUNCTION CODE
000020  016701  000000G                         MOV     $VALUE,R1       ; GET WORD COUNT
000024  001423                                 BEQ     50$             ; IF ZERO, DO NOTHING
000026  060167  000000G                         ADD     R1,$SIZE        ; ADJUST DATA BASE SIZE
000032  060167  000000G                 10$:    ADD     R1,$SIZE
000036  010146                                 MOV     R1,-(SP)        ; SAVE COUNT
000040  012701  000400                         MOV     #256.,R1
000044  021601                         20$:    CMP     (SP),R1         ; IS COUNT =< 256.?
000046  101404                                 BLOS    30$             ; IF LOS, YES
000050                                          CALL    40$             ; DO 256. WORTH
000054  160116                                 SUB     R1,(SP)         ; REDUCE COUNT BY 256.
000056  000772                                 BR      20$
000060  012601                         30$:    MOV     (SP)+,R1
000062                                  40$:    CALL    $RQ2            ; REQUEST BINARY FILE
000066  103402                                 BCS     50$             ; IF CS, NO ACCESS
000070  110022                                 MOVB    R0,(R2)+        ; STORE FUNCTION CODE
000072  110112                                 MOVB    R1,(R2)         ; STORE BYTE/WORD COUNT
                                                                        ; NOTE - 256. IS STORED AS 0
000074                                  50$:    RETURN

                                                .DSABL  LSB
```

```
                                        ;+
                                        ; $ADDRW - ".ADDRW" ACTION ROUTINE
                                        ; $CTM - ".CTIM" ACTION ROUTINE
                                        ; $SLNW - ".SLNW" ACTION ROUTINE
                                        ; $STNW - ".STNW" ACTION ROUTINE
                                        ; $UNW - ".UNW" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2=DESTROYED
                                        ;-

                                                .ENABL  LSB

000654                                  $ADDRW::
000654  012701  000000G                        MOV     #.ADDRW,R1      ; SET FUNCTION CODE
000660  000413                                 BR      10$

000662  012701  000000G                 $CTM::  MOV     #.CTIM,R1       ; SET FUNCTION CODE
000666  000412                                 BR      15$

000670                                  $STNW::
000670  012701  000000G                        MOV     #.STNW,R1       ; SET FUNCTION CODE
000674  000405                                 BR      10$
000676                                  $SLNW::
000676  012701  000000G                        MOV     #.SLNW,R1       ; SET FUNCTION CODE
000702  000402                                 BR      10$
000704                                  $UNW::
000704  012701  000000G                        MOV     #.UNW,R1        ; SET FUNCTION CODE
000710                                  10$:    CALL    $VXLLC          ; MAKE SURE IT'S NOT AN LLC
000714                                  15$:    CALL    $RQ1            ; REQUEST BINARY FILE
000720  103404                                 BCS     20$             ; IF CS, NO ACCESS
000722  110112                                 MOVB    R1,(R2)         ; STORE FUNCTION CODE
000724  062767  000002  000000G                ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
000732                                  20$:    RETURN

                                                .DSABL  LSB
```

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$LFLHD	000416 RG	#14-294
$LSTHD	000454 RG	#14-303
$LTAB	000476 RG	#15-321
$MOD	001260 RG	#24-592
$MPLD	= ****** GX	*14-300
$MPLHD	000432 RG	#14-298
$NOD	001064 RG	#22-526
$NOMOD	001342 RG	#25-616
$NUM	001154 RG	#23-560
$ODD	000520 RG	#16-337
$OFF	= ****** GX	*13-264
$PECHA	000256 RG	#11-226
$PRI	000556 RG	#17-358
$PRI1	000552 RG	#17-357
$PROV	001106 RG	#22-532
$REP.C	= ****** GX	5-74 5-75 5-76 5-77 5-78 5-79 5-80 5-81
$RQ1	= ****** GX	10-211 11-228 14-304 15-322 17-360 18-394 19-434 20-459 20-485 22-526
$RQ2	= ****** GX	6-129 7-151 9-193 16-340 22-532
$RQ3	= ****** GX	21-506
$SIZE	= ****** GX	*6-119 *6-120 *7-155 *9-197 *10-214 *11-231 12-247 13-270 14-295 14-300 *14-307 *15-325 16-338 *16-344 *17-363 *18-397 *19-437 *20-462 *20-488 *21-511 *22-529 *22-540
$SLNB	000620 RG	#18-388
$SLNW	000676 RG	#19-428
$SLTA	= ****** GX	8-170 8-176
$STNB	000612 RG	#18-385
$STNW	000670 RG	#19-425
$SZPTR	= ****** GX	12-248
$TERR	= ****** GX	5-74 5-75 5-76 5-79 5-80 5-81
$TIME	001002 RG	#20-483
$TWARN	= ****** GX	5-77 5-78
$UNB	000626 RG	#18-391
$UNW	000704 RG	#19-431
$VALUE	= ****** GX	6-112 6-117 *7-149 7-154 *8-171 *8-179 12-249 13-264 13-270 *21-504 21-509 21-510 22-534 22-536 22-539 *23-561 23-566 *23-570 *23-574
$VFDDM	= ****** GX	8-169 8-175 9-192 17-359
$VFDLC	= ****** GX	14-299 20-458 20-480
$VFLV1	= ****** GX	13-265
$VFSPC	= ****** GX	14-302
$VXLLC	001434 R	10-210 11-227 18-393 19-433 #27-661
$WORD	001032 RG	8-180 #21-505
$ZBYTE	000076 RG	#7-148
$ZWORD	001026 RG	#21-503
$$PRI	= ****** GX	*17-357
.ADDRB	= ****** GX	18-382
.ADDRW	= ****** GX	19-419
.APR	= ****** GX	20-454
.BLKB	= ****** GX	6-111
.BLKW	= ****** GX	6-116
.BYTE	= ****** GX	7-153 16-342

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 4

```
                                        .SBTTL  $CLIN - CLEAR LINE

                                ;+
                                ; $CLIN - CLEAR LINE
                                ;
                                ; THIS ROUTINE IS CALLED TO DE-ALLOCATE THE DLC AND DDM LINE TABLES AND TO
                                ; REMOVE THE ASSOCIATED PROCESSES FROM CORE IF THEY HAVE NOT BEEN BROUGHT IN
                                ; VIA "SET PROCESS" AND IF THIS IS THE LAST SLT REFERENCE TO THE PROCESS.
                                ; IF THE LAST LINE ON A KMC HAS BEEN CLEARED, THAT KMC WILL BE REMOVED
                                ; FROM THE "LOADED KMC'S" LIST.
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       LF.RDY!LF.ACT CLEARED FROM L.FLG
                                ;       L.DDS AND L.DLS ZEROED
                                ;-
001102                          $CLIN::
001102  016705  000000G         XLIN:   MOV     $SLTA,R5        ; GET SLT ADDRESS
001106  042765  140000  000000          BIC     #LF.RDY!LF.ACT,L.FLG(R5) ; MARK LINE AS NOT READY AND INACTIVE
001114                                  CALL    SCAN            ; SCAN SLT FOR OTHER ACTIVITY
001120  116504  000002                  MOVB    L.DDM(R5),R4    ; GET DDM PDV INDEX
001124                                  CALL    GETPDV          ; GET PDV ADDRESS
001130  032764  000000G 000000G         BIT     #ZF.PSE,Z.FLG(R4) ; IS THIS DLM?
001136  001062                          BNE     20$             ; BR IF YES
001140  005702                          TST     R2              ; ARE ANY OTHER MUX LINES STILL READY?
001142  001054                          BNE     10$             ; IF NE, YES
001144  016503  000016                  MOV     L.KRBA(R5),R3   ; GET KRB ADDRESS
001150  001516                          BEQ     101$            ; IF ZERO, VERY BAD

                                        .IF DF  R$$MPL
                                        TSTB    @NCPU           ; IS THIS A MULTI-PROCESSOR SYSTEM ?
                                        BEQ     1$              ; IF EQ, NO
                                        MOV     K.URM(R3),SETAF+$.AFAF ; GET UNIBUS RUN MASK FOR DEVICE
                                        DIR$    #SETAF          ; SET AFFINITY TO DEVICE'S BUS
                                1$:
                                        .ENDC   ; DF R$$MPL

001152  116303  000001                  MOVB    K.VCT(R3),R3    ; GET VECTOR ADDRESS (ASSUME < 1000)
001156  006303                          ASL     R3              ; CONVERT TO REAL VECTOR ADDRESS
001160  006303                          ASL     R3              ; ...
001162                                  CALL    FNDADD          ; FIND LINE TABLE ADDRESS

                                        .IF DF  R$$MPL
                                        TSTB    @NCPU           ; IS THIS A MULTIPROCESSOR SYSTEM ?
                                        BEQ     3$              ; IF EQ, NO
                                        RMAF$S                  ; ELSE, REMOVE PROCESSOR AFFINITY
                                3$:
                                        .ENDC   ; DF R$$MPL

001166  116504  000002                  MOVB    L.DDM(R5),R4    ; GET THE DDM PDV INDEX
001172                                  CALL    FL116           ; FIND LINE TABLE IN UNLOAD DATA

                                        .IF DF  M$$MGE
                                ;
                                ; FIND THE PDV BIAS
```

```
                                56$:
                                
002512  017700  000000G                MOV     @DECPT,R0       ; GET DECNET HOME BLOCK ADDRESS
002516  001404                        BEQ     47$             ; IF NOT ALLOCATED BRANCH
002520  012701  000104                MOV     #D$END,R1       ; GET BLOCK SIZE
002524                                CALL    $DEA16          ; DEALLOCATE HOME BLOCK
                                
002530                          47$:
                                
002530  005077  000000G                CLR     @CXOPT          ; NO COMM EXEC OPTIONS SET NOW
                                
                                        .IF DF  R$$11M
                                      .IF DF  M$$MGE ! R$$11M
                                    .IF DF  R$$MPL
                                
                                        SWSTK$  60$             ;; ENTER KERNEL MODE
                                        MOV     @NTLPT,R0       ;; GET DATA BLOCK BIAS
                                        CALL    $MAPX           ;; MAP TO THE DATA BLOCK
                                        MOV     .CXALL(R2),R0   ;; GET THE CETAB ALLOCATION ADDRESS
                                        MOV     .CXLEN(R2),R1   ;; AND LENGTH OF ALLOCATION
                                        MOV     R2,-(SP)        ;; SAVE BASE ADDRESS
                                        CALL    $DECEX          ;; DE-ALLOCATE THE CETAB ALLOCATION
                                        MOV     (SP)+,R2        ;; RESTORE BASE ADDRESS
                                
                                        .IFF    ; DF R$$MPL
                                
002534                                  SWSTK$  57$             ;; ENTER KERNEL MODE
002540  017700  000000G                 MOV     @NTLPT,R0       ;; GET DATA BLOCK BIAS
002544                                  CALL    $MAPX           ;; MAP TO THE DATA BLOCK
002550  016266  000000G 000004          MOV     .CXALL(R2),4(SP) ;; GET THE CETAB ALLOCATION ADDRESS
002556  016266  000000G 000006          MOV     .CXLEN(R2),6(SP) ;; AND LENGTH OF ALLOCATION
002564                                  RETURN                  ;; RETURN TO USER MODE
002566                          57$:    CALL    $DEA16          ; DE-ALLOCATE THE CETAB ALLOCATION
                                
002572                                  SWSTK$  60$             ;; ENTER KERNEL MODE
002576  017700  000000G                 MOV     @NTLPT,R0       ;; GET DATA BLOCK BIAS
002602                                  CALL    $MAPX           ;; MAP TO THE DATA BLOCK
002606  016266  000000G 000006          MOV     .CXPCB(R2),6(SP) ;; RETURN CEX PCB IN USER R1
                                
                                        .ENDC   ; DF R$$MPL
                                                                ;; REMEMBER - $MAPX ADDS 1 WORD TO STACK
002614  012266  000004                  MOV     (R2)+,4(SP)     ;; NTPOOL PCB ADDRESS IN USER R0
002620  011266  000010                  MOV     (R2),10(SP)     ;; AND NTPOOL UMR BLOCK IN USER R2
002624                                  RETURN                  ;; BACK TO USER MODE
                                          .IFF  ; DF M$$MGE
                                        MOV     @NTLPT,R2       ; GET DATA BLOCK ADDRESS
                                        MOV     .CXALL(R2),R0   ; GET CETAB ALLOCATION ADDRESS
                                        MOV     .CXLEN(R2),R1   ; AND LENGTH OF CETAB ALLOCATION
                                        MOV     R2,-(SP)        ; SAVE R2
                                        CALL    $DEA16          ; DE-ALLOCATE CETAB ALLOCATION
                                        MOV     (SP)+,R2        ; RESTORE R2
                                        MOV     (R2),R0         ; GET NTPOOL PCB ADDRESS
                                        MOV     .CXPCB(R2),R1   ; AND CEX PCB ADDRESS
                                           ENDC  ; DF M$$MGE
                                
                                60$:
002626                                  CALL    $XPAR           ; RELEASE NTPOOL PARTITION
```

```
                                        .SBTTL  XLT16 - DE-ALLOCATE A 16 BIT ALLOCATION LINE TABLE

                                ;+
                                ; XLT16 - DE-ALLOCATE A 16 BIT ALLOCATION LINE TABLE
                                ;
                                ; INPUTS:
                                ;       R3=LINE TABLE ADDRESS
                                ;       R4=PDV INDEX OF DDM/DLC/LLC PROCESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2,R3=DESTROYED
                                ;-

003560                          XLT16:  CALL    FLT16           ; FIND 16 BIT LINE TABLE ADDRESS
003564  016703  000012'         XLT:    MOV     UBA,R3          ; WAS UNLOAD ENTRY FOUND?
003570  001001                          BNE     120$            ; IF NE, YES
003572                                  CRASH                   ; LINE TABLE NOT FOUND IN UNLOAD DATA!!!
003574  042703  177700          120$:   BIC     #^C<77>,R3      ; FIND ENTRY WITHIN LOCAL BLOCK
003600  062703  000014'                  ADD     #BLOCK,R3       ;
003604  010367  000114'                  MOV     R3,UPTR         ; SAVE POINTER TO IT
003610  005023                          CLR     (R3)+           ; SKIP THE PDV INDEX
003612  011300                          MOV     (R3),R0         ; GET LINE TABLE ADDRESS
003614  005023                          CLR     (R3)+           ; ZERO IT (CLEARS C-BIT)
003616  011301                          MOV     (R3),R1         ; GET LINE TABLE SIZE
003620  042701  000003                  BIC     #3,R1           ; CLEAR EXTRANEOUS BITS
003624                                  CALL    .DEA16          ; DE-ALLOCATE LINE TABLE
003630  011346                          MOV     (R3),-(SP)      ; SAVE LINE TABLE SIZE
003632  005023                          CLR     (R3)+           ; ZERO IT
003634  006026                          ROR     (SP)+           ; ANY SUB-ALLOCATIONS?
003636  103047                          BCC     200$            ; IF CC, NO
003640  005046                          CLR     -(SP)           ; CALCULATE ALLOCATION COUNT
003642  005046                          CLR     -(SP)           ; AND LIBRARY COUNT
003644  152316                          BISB    (R3)+,(SP)      ;       ...
003646  152366  000002                  BISB    (R3)+,2(SP)     ;       ...
003652  005063  177776                  CLR     -2(R3)          ; ZERO THE COUNTS
003656  012667  000120'                  MOV     (SP)+,COUNT     ; GET NUMBER OF ALLOCATIONS
003662  001421                          BEQ     180$            ; IF ZERO, NO WORK TO DO
003664  011300                  150$:   MOV     (R3),R0         ; GET ALLOCATION ADDRESS
003666  005023                          CLR     (R3)+           ; ZERO IT
003670  011301                          MOV     (R3),R1         ; GET ALLOCATION SIZE
003672  005023                          CLR     (R3)+           ; ZERO IT
003674  032701  000001                  BIT     #1,R1           ; 16 BIT ALLOCATION?
003700  001003                          BNE     160$            ; IF NE, NO
003702                                  CALL    .DEA16          ; DE-ALLOCATE SUB-ALLOCATION
003706  000404                          BR      170$
003710  000241                  160$:   CLC                     ; DISCARD 16/18 BIT FLAG
003712  006001                          ROR     R1              ;
003714                                  CALL    $DEA18          ; DE-ALLOCATE SUB-ALLOCATION
003720  005367  000120'          170$:   DEC     COUNT           ; DO IT <ALLOCATION COUNT> TIMES
003724  001357                          BNE     150$            ;
003726  012667  000120'          180$:   MOV     (SP)+,COUNT     ; GET NUMBER OF LIBRARIES
003732  001411                          BEQ     200$            ; IF ZERO, NO WORK TO DO
003734  011300                  190$:   MOV     (R3),R0         ; GET LIBRARY BLOCK BIAS
003736  005023                          CLR     (R3)+           ; ZERO IT
003740  011302                          MOV     (R3),R2         ; GET LIBRARY BLOCK ADDRESS
003742  005023                          CLR     (R3)+           ; ZERO IT
003744                                  CALL    $XLBR           ; RELEASE LIBRARY REFERENCE
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
AC$DNT	= 000002	#5-66							
AC$X25	= 000001	#5-66							
AE$CIR	= 000003	#5-66							
AE$LIN	= 000001	#5-66							
AE$MOD	= 000004	#5-66							
A.ACC	000020	#5-63	24-1485						
A.LNK	000000	#5-63							
A.NAM	000002	#5-63							
A.PUD	000010	#5-63							
A.REM	000012	#5-63							
A.UCB	000010	#5-63							
BLKSIZ	= 000001	#7-197	28-1717						
BLOCK	000014 R	#7-153	15-703	27-1640	28-1669	28-1710	28-1712	28-1732	28-1747	28-1749
		28-1772							
CCBLH	= ****** GX	19-1127	19-1129						
CCBSZ	= ****** GX	19-1130							
CLKHD	= ****** GX	13-534	19-1078						
CL$ASZ	= 010500	#5-66							
CL$DLL	= 000500	#5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66
		5-66	5-66							
CL$ECL	= 000300	#5-66	5-66							
CL$LDN	= 010400	#5-66	5-66							
CL$MAN	= 000000	#5-66	5-66	5-66	5-66					
CL$PAZ	= 034100	#5-66	5-66	5-66						
CL$PLH	= 034000	#5-66	5-66	5-66	5-66	5-66				
CL$PLL	= 000600	#5-66								
CL$PRT	= 034200	#5-66								
CL$ROU	= 010000	#5-66	5-66	5-66						
CL$SES	= 000200	#5-66	5-66	5-66						
CL$SGE	= 035000	#5-66	5-66	5-66						
CL$SSE	= 035100	#5-66	5-66	5-66	5-66	5-66				
CL$TRN	= 000400	#5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66
		5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66	5-66
		5-66								
CL$XL2	= 013700	#5-66								
CL$XL3	= 013600	#5-66	5-66							
CL$X2S	= 013500	#5-66	5-66							
COUNT	000120 R	#7-168	*28-1686	*28-1699	*28-1701	*28-1708				
CXOPT	= ****** GX	19-1176	19-1302							
DDMAC	000124 RG	#7-174	14-632							
DEAC2	004302 R	29-1855	#30-1911							
DEAECL	000422 R	8-304	#10-379							
DEALST	004424 RG	23-1456	25-1519	#31-1967	32-2004					
DECPT	= ****** GX	19-1169	24-1473	25-1512	32-2000	32-2020				
DEUMR	= ****** GX	19-1254	19-1331							
DEVHD	= ****** GX	20-1367								
DLCAC	000122 RG	#7-173	14-650							
DL$AST	= 000002	#5-66								
DL$HLT	= 000000	#5-66								
DL$IST	= 000001	#5-66								
DL$MAI	= 000004	#5-66								
DL$OFF	= 000001	#5-66								

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 2

```
                        .TITLE  NTL11M - RSX-11M DEFINITIONS
                        .IDENT  /V05.00/

                ;
                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ; MODULE DESCRIPTION:
                ;
                ;       NTL - RSX11M GLOBAL SYMBOL DEFINITIONS
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  27-FEB-78
                ;       VERSION 2.0 RELEASE
                ;
                ; 2.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                        .SBTTL  LOCAL DATA
                                ;
                                ; LOCAL DATA
                                ;
000000  000003                  .WORD   3
000002                  XAVLL:  .BLKW   1
000004  000000                  .WORD   0
```

Table of contents

5- 56 MACRO DEFINITIONS
6- 64 $ALP16 - ALLOCATE FROM PROCESS PARTITION
7- 157 $ALPRO - ALLOCATE STORAGE FOR PROCESS

```
                                        .SBTTL  LOCAL SYMBOL DEFINTIONS


                                ;
                                ; LOCAL SYMBOLS
                                ;


                                ;
                                ; ERROR CODES RETURNED TO CALLER
                                ;
000000                                  .ASECT
000002                                  .=2                     ; 0=SUCCESS
000002                          ERR2:   .BLKW   1               ; OPEN FAILURE
000004                          ERR4:   .BLKW   1               ; GET ATTRIBUTES FAILED
000006                          ERR6:   .BLKW   1               ; FILE NOT CONTIGUOUS
000010                          ERR8:   .BLKW   1               ; LABEL BLOCK READ FAILURE
000012                          ERR10:  .BLKW   1               ; INVALID PROCESS IMAGE
000014                          ERR12:  .BLKW   1               ; PROCESS NOT BUILT FOR APR5 (MAPPED)
000016                          ERR14:  .BLKW   1               ; PROCESS > 4K WORDS (MAPPED)
000020                          ERR16:  .BLKW   1               ; PROCESS EXTENSION > 4K WORDS (MAPPED)
000022                          ERR18:  .BLKW   1               ; PARTITION NOT IN SYSTEM
000024                          ERR20:  .BLKW   1               ; PROCESS/PARTITION BASE MISMATCH
000026                          ERR22:  .BLKW   1               ; PARTITION TOO SMALL
000030                          ERR24:  .BLKW   1               ; NOT COMMON PARTITION
000032                          ERR26:  .BLKW   1               ; PARTITION BUSY
000034                          ERR28:  .BLKW   1               ; CEX PARTITION NOT IN EXEC SPACE (MAPPED)
000036                          ERR30:  .BLKW   1               ; SUB-PCB ALLOCATION FAILURE (MAPPED)
000040                          ERR32:  .BLKW   1               ; MAIN PARTITION TOO FRAGMENTED (MAPPED)
000042                          ERR34:  .BLKW   1               ; PROCESS READ FAILURE
000044                          ERR36:  .BLKW   1               ; FAILED TO CREATE PROCESS REGION (IAS ONLY)
000046                          ERR38:  .BLKW   1               ; FAILED TO MAP TO PROCESS REGION (IAS ONLY)
000000                                  .PSECT
```

```
633 001452 010116                          MOV     R1,(SP)          ;* SAVE PREVIOUS SUB-PCB ADDRESS
634 001454 010260 000014                   MOV     R2,P.REL(R0)     ;* SET SUB PARTITION BIAS
635 001460 010360 000010                   MOV     R3,P.SUB(R0)     ;* AND POINTER TO NEXT SUB-PCB IN LIST
636 001464 032767 000000G 000000G          BIT     #FL.TOP,$FLAGS   ;* TOP-DOWN ALLOCATION?
637 001472 001430                          BEQ     50$              ;* IF EQ, NO - ALL DONE
638 001474 060460 000014                   ADD     R4,P.REL(R0)     ;* ALLOCATE FROM TOP OF HOLE
639 001500 010301                  20$:    MOV     R3,R1            ;* COPY SUB-PCB ADDRESS
640 001502 016102 000016                   MOV     P.BLKS(R1),R2    ;* GET SUB-PCB SIZE
641 001506 000746                          BR      10$              ;*
642 001510 016504 000014           30$:    MOV     P.REL(R5),R4     ;* GET MAIN PARTITION BIAS
643 001514 066504 000016                   ADD     P.BLKS(R5),R4    ;* GIVING BIAS OF END OF PARTITION
644 001520 160204                          SUB     R2,R4            ;* GIVING SIZE OF LAST HOLE
645 001522 166704 000000'                  SUB     BLOX,R4          ;* IS LAST HOLE BIG ENOUGH?
646 001526 103412                          BLO     50$              ;* IF MI, NO
647 001530 032767 000000G 000000G          BIT     #FL.TOP,$FLAGS   ;* TOP-DOWN ALLOCATION?
648 001536 001401                          BEQ     40$              ;* IF EQ, NO
649 001540 060402                          ADD     R4,R2            ;* ALLOCATE FROM TOP OF HOLE
650 001542 010116                  40$:    MOV     R1,(SP)          ;* SAVE PREVIOUS SUB-PCB ADDRESS
651 001544 010260 000014                   MOV     R2,P.REL(R0)     ;* SET SUB PARTITION BIAS
652 001550 010360 000010                   MOV     R3,P.SUB(R0)     ;* AND POINTER TO NEXT SUB-PCB IN LIST
653 001554 012601                  50$:    MOV     (SP)+,R1         ;* GET PREVIOUS SUB-PCB ADDRESS
654 001556 001410                          BEQ     111$             ;* IF EQ, NO SPACE IN PARTITION
655 001560 010061 000010                   MOV     R0,P.SUB(R1)     ;* LINK OUR SUB-PCB TO PREVIOUS
656 001564 010005                          MOV     R0,R5            ;* THIS IS NOW THE OFFICIAL PCB
657 001566                                 RETURN                   ;*
658
659                                        ;
660                                        ; ERROR CONDITIONS
661                                        ;
662 001570 012767 000036 000010'   101$:   MOV     #ERR30,ERROR     ;* SUB-PCB ALLOCATION FAILURE
663 001576                                 RETURN                   ;*
664 001600 005005                  111$:   CLR     R5               ;* INIDICATE NO SUB-PARTITION ALLOCATED
665 001602 012767 000040 000010'           MOV     #ERR32,ERROR     ;* MAIN PARTITION TOO FRAGMENTED
666 001610 016701 000000G                  MOV     PLGTH,R1         ;* GET PCB LENGTH
667 001614                                 CALLR   @DEACB           ;* DE-ALLOCATE SUB-PCB
668
669                                         .ENDC
670                                        .IFF    ; DF R$$11M
```

```
                                        .SBTTL  LOCAL DATA

                                ; ****
                                ; DEFINE LISTHEAD FOR PROCESS ALLOCATIONS
                                ; ****

                                        .IF DF  R$$11M

000000  000003                          .WORD   3                       ;ROUNDING FACTOR
000002  000000          $PRAVL::.WORD   0                       ;ADDRESS OF START OF POOL
000004  000000                          .WORD   0                       ; NEEDED FOR DEALLOCATIONS

                                        .ENDC   ; DF R$$11M

                                        .IIF NDF M$$MGE .END
```

```
                                        .SBTTL  PERR - PRINT ERROR MESSAGE
                                ;+
                                ; PERR - PRINT ERROR MESSAGE
                                ;
                                ; INPUTS:
                                ;       R0=MESSAGE BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SET
                                ;       R0,R1=DESTROYED
                                ;-
000000                          PERR:   CALL    $SAVAL          ; SAVE ALL REGISTERS
000004  010005                          MOV     R0,R5           ; COPY ERROR MESSAGE BLOCK ADDRESS
000006  012704  002272'                 MOV     #ERRBUF,R4      ; PLACE TO BUILD ERROR MESSAGE
000012  012503                          MOV     (R5)+,R3        ; GET FORMAT STRING ADDRESS
000014                                  CALL    @(R5)+          ; DO THE FORMATTING
000016  105713                          TSTB    (R3)            ; IS IT END OF FORMAT STRING?
000020  001403                          BEQ     40$             ; IF EQ, YES
000022  112324                  30$:    MOVB    (R3)+,(R4)+     ; ELSE COPY REST OF FORMAT STRING
000024  001376                          BNE     30$             ;
000026  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000030  012503                  40$:    MOV     (R5)+,R3        ; GET LINE 2 ADDRESS
000032  112524                  50$:    MOVB    (R5)+,(R4)+     ; COPY THE ERROR TEXT
000034  001376                          BNE     50$             ;
000036  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000040  012700  002262'                 MOV     #ERRMSG,R0      ; SET ERROR MESSAGE ADDRESS
000044  010401                          MOV     R4,R1           ; COPY END ADDRESS
000046  160001                          SUB     R0,R1           ; GIVING LENGTH
000050                                  CALL    $TIOUT          ; PRINT IT
000054  005703                          TST     R3              ; SHOULD SECOND LINE BE PRINTED?
000056  001401                          BEQ     60$             ; IF EQ, NO
000060                                  CALL    (R3)            ; PRINT SECOND ERROR MESSAGE LINE
000062  000261                  60$:    SEC                     ; RETURN WITH C-BIT SET
000064                          RTSPC:  RETURN
```

```
    134 000170                          50$:    RETURN
    135
```

```
                        ;***
                        ; LOCAL SYMBOLS
                        ;***


                        ;
                        ; ERROR CODES RETURNED TO CALLER
                        ;
                                .ASECT
000002                          . = 2
000002  E.OPEN: .BLKW   1               ; 0 = SUCCESS
000004  E.READ: .BLKW   1               ; 2 = OPEN FAILURE
000006  E.ORUN: .BLKW   1               ; 4 = READ FAILURE
000010  E.BAD:  .BLKW   1               ; 6 = RECORD BUFFER TOO SMALL
000000                                  ; 8. = BAD FILE FORMAT (INCLUDES PREMATURE EOF)
                                .PSECT


                        ;
                        ; SYMBOL BLOCK LAYOUT
                        ;
                                .ASECT
000000                          . = 0
000000  SB.LNK: .BLKW   1               ; LINK TO NEXT SYMBOL BLOCK (0=LAST)
000002  SB.DEF: .BLKW   1               ; NON-ZERO IF SYMBOL IS DEFINED
000004  SB.SYM: .BLKW   2               ; RAD50 SYMBOL
000010  SB.VAL: .BLKW   1               ; SYMBOL VALUE
000000                                  .PSECT

                        ;***
                        ; LOCAL DATA
                        ;***

000000                          .PSECT  DATA,D

                        ;
                        ; SYMBOL TABLE RECORD BUFFER
                        ;
000000  STBBF:  .BLKB   124.

000000                          .PSECT
```

```
                                        ;+
                                        ; $BYTE - ".BYTE" ACTION ROUTINE
                                        ; $ZBYTE - ACTION ROUTINE TO STORE A ZERO BYTE
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=BYTE VALUE
                                        ;
                                        ; OUTPUTS:
                                        ;       R2=DESTROYED
                                        ;-
000076                                  $ZBYTE::
000076  005067  000000G                         CLR     $VALUE          ; USE ZERO AS EXPRESSION VALUE
000102                                  $BYTE::
000102                                          CALL    $RQ2            ; REQUEST BINARY FILE
000106  103406                                  BCS     10$             ; IF CS, NO ACCESS
000110  112722  000000G                         MOVB    #.BYTE,(R2)+    ; STORE FUNCTION CODE
000114  116712  000000G                         MOVB    $VALUE,(R2)     ; STORE BYTE VALUE
000120  005267  000000G                         INC     $SIZE           ; ADJUST DATA BASE SIZE
000124                                  10$:    RETURN
```

```
                                        ;+
                                        ; $APR - ".APR" ACTION ROUTINE
                                        ; $DPRB - ".DPRB" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2=DESTROYED
                                        ;-
                                                .ENABL  LSB
000734                                  $APR::
000734  012701  000000G                         MOV     #.APR,R1        ; FUNCTION CODE
000740  000402                                  BR      10$             ; CONTIUE IN COMMON CODE
000742                                  $DPRB::
000742  012701  000000G                         MOV     #.DPRB,R1       ; FUNCTION CODE
000746                                  10$:    CALL    $VFDLC          ; MAKE SURE IT'S A DLC
000752                                          CALL    $RQ1            ; REQUEST BINARY FILE
000756  103403                                 BCS     20$             ; IF CS, NO ACCESS
000760  110112                                 MOVB    R1,(R2)         ; STORE FUNCTION CODE
000762  005267  000000G                         INC     $SIZE           ; ADJUST DATA BASE SIZE
000766                                  20$:    RETURN

                                                .DSABL  LSB


                                        ;+
                                        ; $DPRW - ".DPRW" ACTION ROUTINE
                                        ; $TIME - ".TIME" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R1,R2=DESTROYED
                                        ;-

                                                .ENABL  LSB
000770                                  $DPRW::
000770                                          CALL    $VFDLC          ; MAKE SURE THIS IS A DLC
000774  012701  000000G                         MOV     #.DPRW,R1       ; SET FUNCTION CODE
001000  000402                                  BR      10$
001002                                  $TIME::
001002  012701  000000G                         MOV     #.TIME,R1       ; SET FUNCTION CODE
001006                                  10$:    CALL    $RQ1            ; REQUEST BINARY FILE
001012  103404                                 BCS     20$             ; IF CS, NO ACCESS
001014  110112                                 MOVB    R1,(R2)         ; STORE FUNCTION CODE
001016  062767  000002  000000G                ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
001024                                  20$:    RETURN

                                                .DSABL  LSB
```

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
.CSR	= ******	GX	9-195		
.CTIM	= ******	GX	19-422		
.DPRB	= ******	GX	20-457		
.DPRW	= ******	GX	20-481		
.DVCHA	= ******	GX	10-213		
.LSTHD	= ******	GX	14-306		
.LTAB	= ******	GX	15-324		
.NODE	= ******	GX	22-528		
.PECHA	= ******	GX	11-230		
.PNUMB	= ******	GX	23-561	24-593	*24-601
.PNUMH	= ******	GX	23-562	24-594	
.PRI	= ******	GX	17-362		
.PRTOV	= ******	GX	22-538		
.PSTCN	= ******	GX	26-635		
.PSTPT	= ******	GX	26-633		
.SLNB	= ******	GX	18-389		
.SLNW	= ******	GX	19-429		
.STNB	= ******	GX	18-386		
.STNW	= ******	GX	19-426		
.TIME	= ******	GX	20-484		
.UNB	= ******	GX	18-392		
.UNW	= ******	GX	19-432		
.WORD	= ******	GX	21-508		

```
                                        ;
001176                                          CALL    GETPDV          ; CONVERT INDEX TO PDV ADDRESS
001202  010467  000152'                        MOV     R4,..PDV        ; SAVE PDV INDEX
001206  016467  000000G 000000G                MOV     Z.DSP(R4),$BIAS ; GET THE PDV BIAS

                                                .IF DF  R$$MPL
                                                CALL    FNCTB           ; FIND THE CTB ADDRESS
                                                MOVB    L.CTL(R5),R0    ; GET THE CONTROLLER NUMBER
                                                ASL     R0              ; CONVERT TO A WORD INDEX
                                                ADD     R0,R1           ; GET ADDRESS OF KRB IN CTB
                                                CLR     L.KRB(R1)       ; ZERO KRB POINTER
                                                CALL    $HRCIN          ; RE-INITIALIZE HRC...
                                                .ENDC

                                                .ENDC   ; DF M$$MGE

001214                                          CALL    VECT            ; RELEASE VECTOR(S)
                                        ;
                                        ; DE-ALLOCATE KRB ALLOCATION
                                        ;
001220  016500  000016                          MOV     L.KRBA(R5),R0   ; GET KRB ADDRESS
001224                                          CALL    XKRB            ; DE-ALLOCATE KRB

001230                                          CALL    XLT             ; DE-ALLOCATE LINE TABLE
001234                                          CALL    ZDDS            ; ZERO L.DDS IN SLT(S)
001240  016704  000152'                         MOV     ..PDV,R4        ; GET PDV ADDRESS
001244  016467  000000G 000000G                MOV     Z.NAM(R4),$NAME ; SETUP DDM DEVICE NAME
001252  005767  000124'          5$:   TST     DDMAC                   ; ANY OTHER DDM ACTIVITY?
001256  001006                                  BNE     10$             ; IF NE, YES
001260  132764  000000G 000000G                BITB    #ZF.MFL,Z.FLG(R4) ; IS DDM PROCESS MARKED FOR LOAD?
001266  001002                                  BNE     10$             ; IF NE, YES - DON'T REMOVE
001270                                          CALL    XPRO            ; REMOVE PROCESS FROM CORE
001274  126565  000002  000003  10$:   CMPB    L.DDM(R5),L.DLC(R5) ; COMBINATION DDM/DLC?
001302  001434                                  BEQ     30$             ; IF EQ, YES - ALL DONE
001304  016503  000010          20$:   MOV     L.DLS(R5),R3    ; GET DLC LINE TABLE ADDRESS
001310  001437                                  BEQ     111$            ; IF ZERO, VERY BAD

001312  116504  000003                          MOVB    L.DLC(R5),R4    ; GET DLC PDV INDEX
001316                                          CALL    GETPDV          ; CONVERT INDEX TO PDV ADDRESS
001322  010467  000152'                         MOV     R4,..PDV        ; DON'T DISCARD PARTITION LINE TABLES

001326  116504  000003                          MOVB    L.DLC(R5),R4    ; GET DLC PDV INDEX
001332                                          CALL    XLT16           ; DE-ALLOCATE DLC LINE TABLE
001336  005065  000010                          CLR     L.DLS(R5)       ; ZERO LINE TABLE ADDRESS
001342  005065  000006                          CLR     L.DLM(R5)       ; AND LINE TABLE BIAS
001346  005767  000122'                         TST     DLCAC           ; ANY OTHER DLC ACTIVITY?
001352  001010                                  BNE     30$             ; IF NE, YES
001354  016704  000152'                         MOV     ..PDV,R4        ; GET PDV ADDRESS
001360  132764  000000G 000000G                BITB    #ZF.MFL,Z.FLG(R4) ; IS DLC PROCESS MARKED FOR LOAD?
001366  001002                                  BNE     30$             ; IF NE, YES - DON'T REMOVE
001370                                          CALL    XPRO            ; REMOVE DLC FROM CORE
001374  042765  020000  000000  30$:   BIC     #LF.UNL,L.FLG(R5) ; CLEAR MARKED FOR UNLOAD FLAG
001402  000241                                  CLC                     ; RETURN C-BIT CLEAR
001404                                          RETURN

                                        ;
                                        ; ERROR CONDITIONS
```

```
                                        BCS     101$            ; IF CS, VERY BAD

                                            .IFF    ; DF R$$11M
                                        MOV     @NTLPT,R0       ; GET DATA BLOCK BIAS
                                        CALL    $MAPX           ; MAP TO THE DATA BLOCK
                                        MOV     (R2),R0         ; GET GCD ADDRESS OF NTPOOL
                                        CALL    @(SP)+          ; CALL CO-ROUTINE TO RESTORE MAPPING
                                        CALL    $ULOCK          ; UNLOCK THE PROCESS
                                            .ENDC   ; DF R$$11M

                                        .IF NDF R$$MPL

002634  010100                         MOV     R1,R0           ; GET CEX PCB ADDRESS
002636                                  CALL    $XPAR           ; RELEASE CEX FROM ITS PARTITION

                                        .ENDC   ; NDF R$$MPL

002642  042777  000000C 00C000G        BIC     #FE.CEX,@FMASK  ; INDICATE COMM EXEC NOT IN MEMORY

                                        .IF DF  R$$MPL
                                        MOVB    @NCPU,R4        ; GET NUMBER OF PROCESSORS IN SYSTEM
                                        BEQ     67$             ; IF EQ, NOT A MULTIPROCESSOR SYSTEM
                                        CALL    XUMR            ; RELEASE UMR'S
                                        BR      70$             ; AND CONTINUE
                                67$:
                                        .ENDC

002650  010200                         MOV     R2,R0           ; COPY UMR BLOCK ADDRESS
002652  001410                         BEQ     70$             ; IF ZERO, NO UMR'S TO WORRY ABOUT
002654                                  SWSTK$  70$             ;; ENTER KERNEL MODE
002660                                  CALL    @DEUMR          ;; DE-ASSIGN UMR'S
002664  012701  000000G                 MOV     #M.LGTH,R1      ;; GET UMR BLOCK LENGTH
002670                                  CALLR   $DECEX          ;; DE-ALLOCATE UMR BLOCK

002674  000241                  70$:    CLC                     ; INDICATE SUCCESS
002676                                  RETURN

                                ;
                                ; "SET CEX" ERROR RECOV
                                ;

002700                          80$:
                                        .IF DF  R$$11M
                                        .IF DF  M$$MGE

002700  017700  000000G                 MOV     @NTLPT,R0       ; GET THE NTL HOME BLOCK BIAS
002704  001455                         BEQ     90$             ; IF EQ, THERE IS NONE
002706                                  SWSTK$  83$             ;; ENTER KERNEL MODE
002712                                  CALL    $MAPX           ;; MAP TO NTL HOME BLOCK
002716  016266  000000G 000004          MOV     .CXALL(R2),4(SP) ;; RETURN CETAB ALLOCATION ADDRESS IN R0
002724  016266  000000G 000006          MOV     .CXLEN(R2),6(SP) ;; RETURN CETAB LENGTH IN R1
                                                                ;; REMEMBER $MAPX ADD 1 WORD TO SP
002732                                  RETURN                  ;; RETURN TO USER MODE

                                        .IFF

                                        CLR     R0              ; ASSUME NO CETAB ALLOCATION
```

```
1708 003750 005367 000120'                  DEC     COUNT           ; DO IT <LIBRARY COUNT> TIMES
1709 003754 001367                          BNE     190$            ;       ...
1710 003756 022767 000016' 000114' 200$:    CMP     #BLOCK+2,UPTR   ; WAS OUR LINE TABLE THE FIRST ENTRY?
1711 003764 001043                          BNE     250$            ; IF NE, NO
1712 003766 022703 000114'                  CMP     #BLOCK+100,R3   ; DID IT FILL THE ENTIRE 32. WORD BLOCK?
1713 003772 001402                          BEQ     210$            ; IF EQ, YES
1714 003774 005713                          TST     (R3)            ; ANYTHING SIGNIFICANT IN REST OF BLOCK?
1715 003776 001036                          BNE     250$            ; IF NE, YES
1716 004000 016700 000010'          210$:   MOV     UBIAS,R0        ; GET UNLOAD BLOCK ADDRESS
1717 004004 012701 000001                   MOV     #BLKSIZ,R1      ; AND SIZE
1718 004010                                 CALL    $DEA18          ; DE-ALLOCATE UNLOAD BLOCK
1719
1720                                        .IF DF  M$$MGE
1721 004014 017700 000000G                  MOV     @NTLPT,R0       ; GET NETLDR DATA BLOCK BIAS
1722                                        .IF DF  R$$11M
1723 004020                                 SWSTK$  300$            ;; ENTER KERNEL MODE
1724                                        .ENDC
1725 004024                                 CALL    $MAPX           ;; MAP TO THE DATA BLOCK
1726                                        .IFF
1727                                        MOV     @NTLPT,R2       ; GET NETLDR DATA BLOCK ADDRESS
1728                                        .ENDC
1729
1730 004030 026267 000000G 000010'          CMP     .CXUNL(R2),UBIAS ;; WAS OUR UNLOAD BLOCK THE FIRST?
1731 004036 001004                          BNE     220$            ;; IF NE, NO
1732 004040 016762 000014' 000000G          MOV     BLOCK,.CXUNL(R2) ;; SET NEW FIRST UNLOAD BLOCK ADDRESS
1733 004046 000446                          BR      300$            ;;
1734 004050                         220$:
1735                                        .IF DF  M$$MGE
1736 004050 016211 000000G                  MOV     .CXUNL(R2),(R1) ;; MAP TO THE FIRST UNLOAD BLOCK
1737 004054 000401                          BR      240$            ;;
1738 004056 011211                  230$:   MOV     (R2),(R1)       ;; MAP TO THE NEXT UNLOAD BLOCK
1739                                        .IFF
1740                                        MOV     .CXUNL(R2),R2   ; GET FIRST UNLOAD BLOCK ADDRESS
1741                                        BR      240$
1742                                230$:   MOV     (R2),R2         ; GET NEXT UNLOAD BLOCK ADDRESS
1743                                        .ENDC
1744
1745 004060 021267 000010'          240$:   CMP     (R2),UBIAS      ;; DOES THIS BLOCK POINT TO OURS?
1746 004064 001374                          BNE     230$            ;; IF NE, NO
1747 004066 016712 000014'                  MOV     BLOCK,(R2)      ;; CHANGE NEXT BLOCK POINTER
1748 004072 000434                          BR      300$            ;;
1749 004074 012700 000114'          250$:   MOV     #BLOCK+100,R0   ; GET END OF BLOCK ADDRESS
1750 004100 020003                          CMP     R0,R3           ; ARE WE THERE NOW?
1751 004102 001413                          BEQ     280$            ; IF EQ, YES
1752 004104 010001                          MOV     R0,R1           ; CALCULATE SIZE REMAINING IN BLOCK
1753 004106 160301                          SUB     R3,R1           ;       ...
1754 004110 006201                          ASR     R1              ;       ...
1755 004112 016702 000114'                  MOV     UPTR,R2         ; COPY REST OF BLOCK ONTO WHAT USED TO BE
1756 004116 012322                  260$:   MOV     (R3)+,(R2)+     ; OUR LINE TABLE ENTRY
1757 004120                                 SOB     R1,260$         ;       ...
1758 004124 005022                  270$:   CLR     (R2)+           ; ZERO THE REST OF THE BLOCK
1759 004126 020002                          CMP     R0,R2           ;       ...
1760 004130 001375                          BNE     270$            ;       ...
1761 004132                         280$:
1762                                        .IF DF  M$$MGE
1763 004132 016700 000010'                  MOV     UBIAS,R0        ; GET UNLOAD BLOCK BIAS
1764                                        .IF DF  R$$11M
```

NTLUNL CREATED BY MACRO ON 29-JUN-85 AT 00:35 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
DL$ON	= 000000	#5-66
DL$RUN	= 000003	#5-66
DL$SHU	= 000002	#5-66
DL$SYN	= 000005	#5-66
DQCLK	000766 R	12-486 #13-516 19-1118
D$ANN	000000	24-1475
D$END	= 000104	19-1171
D$FNB	000034	32-2022
D$RNN	000002	25-1514
D$SER	000032	32-2002
D.DSP	= ****** GX	*20-1375
D.PCB	= ****** GX	20-1372 *20-1376
EF$ACT	= 000001	#5-66
EV$ACF	= 000201	#5-66
EV$ADR	= 000420	#5-66
EV$ADU	= 000417	#5-66
EV$APL	= 000400	#5-66
EV$ARC	= 000421	#5-66
EV$AUC	= 000010	#5-66
EV$AUS	= 000003	#5-66
EV$CDF	= 000520	#5-66
EV$COZ	= 000011	#5-66
EV$DBR	= 000302	#5-66
EV$GAS	= 035101	#5-66
EV$HCE	= 035114	#5-66
EV$HCI	= 035113	#5-66
EV$HFE	= 000506	#5-66
EV$IFL	= 000413	#5-66
EV$IFO	= 000415	#5-66
EV$IFS	= 000414	#5-66
EV$INF	= 000515	#5-66
EV$LDL	= 000407	#5-66
EV$LDN	= 010416	#5-66
EV$LDO	= 000411	#5-66
EV$LDS	= 000410	#5-66
EV$LSC	= 000500	#5-66
EV$LUP	= 000412	#5-66
EV$NOL	= 000402	#5-66
EV$NRC	= 000416	#5-66
EV$NSC	= 000200	#5-66
EV$NUL	= 000401	#5-66
EV$NVR	= 000406	#5-66
EV$OPL	= 000403	#5-66
EV$PCC	= 034000	#5-66
EV$PCI	= 034001	#5-66
EV$PCM	= 034002	#5-66
EV$PFE	= 000404	#5-66
EV$PPC	= 034003	#5-66
EV$RCF	= 000517	#5-66
EV$RDC	= 010001	#5-66
EV$RDR	= 010002	#5-66
EV$RJE	= 035106	" 66

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  DCBDF$,HWDDF$,LBLDF$,PCBDF$,SCBDF$,TCBDF$,UCBDF$

                        ;***
                        ; GLOBAL LIBRARY SYMBOLS
                        ;***

                        ;
                        ; DEVICE CONTROL BLOCK OFFSETS (DCB)
                        ;
000000                          DCBDF$
                                        .GLOBL  D.DSP,D.NAM,D.PCB,D.UCB,D.UCBL,D.UNIT,D.VPWF

                        ;
                        ; FEATURES MASK BIT DEFINITIONS
                        ;
000000                          HWDDF$
                                        .GLOBL  FE.CEX,FE.EXT,FE.PLA,FE.FXV
                                        .IIF DF R$$MPL, .GLOBL F2.DAS

                        ;
                        ; TASK LABEL BLOCK OFFSETS
                        ;
000000                          LBLDF$
                                        .GLOBL  L$BEXT,L$BFLG,L$BHRB,L$BLDZ,L$BMXZ,L$BSA,TS$NHD

                        ;
                        ; PARTITION CONTROL BLOCK OFFSETS (PCB)
                        ;
000000                          PCBDF$
                                 .GLOBL PS.COM,PS.NSF,P.MAIN,P.NAM,P.STAT,P.SUB,P.WAIT
                                 .IF NDF R$$MPL
                                .GLOBL  PS.SYS,P.BUSY
                                 .ENDC

                        ;
                        ; STATUS CONTROL BLOCK OFFSETS (SCB)
                        ;
000000                          SCBDF$
                                        .GLOBL  S.CON,S.LHD,M.BFVH,M.BFVL,M.LGTH,M.UMRA,M.UMRN

                        ;
                        ; TASK CONTROL BLOCK OFFSETS (TCB)
                        ;
000000                          TCBDF$
                                        .GLOBL  T.LDV

                        ;
                        ; UNIT CONTROL BLOCK OFFSETS (UCB)
                        ;
000000                          UCBDF$
                                        .GLOBL  UC.PWF,US.MNT,U.CTL,U.DCB,U.RED,U.SCB,U.STS,U.UNIT
                                        .GLOBL  DV.MNT,US.BSY,U.CW1

                        ;***
```

```
                                        .SBTTL  $XALOC - ALLOCATE CORE BUFFER IN EXTENDED POOL
                                ;+
                                ; **-$XALOC-ALLOCATE CORE BUFFER IN EXTENDED POOL
                                ;
                                ; THIS ROUTINE IS CALLED TO ALLOCATE AN CORE BUFFER IN EXTENDED POOL.
                                ; THE ALLOCATION ALGORITHM IS FIRST FIT AND BLOCKS ARE ALLOCATED IN
                                ; MULTIPLES OF FOUR BYTES. IF THE EXTENDED ALLOCATION FAILS, THE
                                ; BLOCK IS ALLOCATED USING RSX POOL.
                                ;
                                ; INPUTS:
                                ;
                                ;       R1=SIZE OF THE CORE BUFFER TO ALLOCATE IN BYTES.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       C=1 IF INSUFFICIENT CORE IS AVAILABLE TO ALLOCATE THE BLOCK.
                                ;       C=0 IF THE BLOCK IS ALLOCATED.
                                ;               R0=ADDRESS OF THE ALLOCATED BLOCK.
                                ;               R1=LENGTH OF BLOCK ALLOCATED
                                ;
                                ; REGISTERS:
                                ;       R2 MODIFIED
                                ;
                                ;-
                                ;
000006                          $XALOC::
                                        .IF NDF M$$MGE
                                ;
                                        CALLR   $ALOCB          ; ALLOCATE FROM RSX POOL IN UNMAPPED SYSTEMS
                                ;
                                        .IFF
                                ;
000006                                  SAVMAP                  ; SAVE CURRENT APR 5 MAPPING
000012  012700  000000'                 MOV     #XAVLL-2,R0     ; POINT TO ALLOCATION MASK WORD
000016  061001                          ADD     (R0),R1         ; ADD ROUNDING FACTOR
000020  042001                          BIC     (R0)+,R1        ; CLEAR EXCESS AND POINT TO LIST HEAD
000022  017710  000000G                 MOV     @XAVL,(R0)      ; COPY CEX POOL LISTHEAD
                                ;
000026  010002                  10$:    MOV     R0,R2           ; SAVE ADDRESS OF CURRENT BLOCK
000030  011200                          MOV     (R2),R0         ; GET ADDRESS OF NEXT BLOCK
000032  001003                          BNE     15$             ; IF NE, ANOTHER BLOCK TO CHECK
                                ;
                                        .IF DF  R$$11M
                                ;
000034                                  CALL    @ALOCB          ; ELSE, TRY TO ALLOCATE FROM RSX POOL
                                ;
                                        .IFF    ; DF R$$11M
                                ;
                                        MOV     R3,-(SP)        ; SAVE SOME
                                        MOV     R4,-(SP)        ;  REGISTERS
                                        MOV     R1,R3           ; COPY SIZE OF BLOCK REQUIRED
                                        CLC                     ; ASSUME SUCCESS
                                ;
                                        .INH0                   ;; INHIBIT TASK SWITCHING
                                        CALL    $$RQCB          ;; ALLOCATE BLOCK OF MEMORY
                                        ADC     (SP)            ;; SET C-BIT FOR RETURN
                                        .ENB0                   ;; ENABLE TASK SWITCHING
```

```
                        .TITLE  PRALL - NTL PROCESS ALLOCATION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;
; MODULE DESCRIPTION:
;
;       CORE ALLOCATION ROUTINES FOR ALLOCATING FROM PROCESS PARTITION
;
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 4.01  27-AUG-84
;       SUPPORT VECTORED EXEC
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
;
;
;
```

```
                                        .SBTTL  LOCAL DATA

                                ;
                                ; LOCAL DATA
                                ;

000000                                  .PSECT  DATA,D

                                ;
                                ; PROCESS SIZES (INCLUDING EXTENSION)
                                ;
000000                          BLOX:   .BLKW   1               ; 32. WORD BLOCKS
000002                          BYTES:  .BLKW   1               ; BYTES
000004  000000                  LNKS:   .WORD   0               ; MAXIMUM LOGICAL LINKS

                                ;
                                ; PROCESS LINE TABLE/CONTROLLER TABLE EXTENSION
                                ;
                                        .IF DF  M$$MGE
000006                          EXTEND: .BLKW   1
                                        .ENDC

                                ;
                                ; ERROR INDICATION
                                ;
000010                          ERROR:  .BLKW   1

                                        .IF DF  R$$11M
                                ;
                                ; PARTITION CONTROL BLOCK ADDRESS
                                ;
000012                          PCBA:   .BLKW   1

                                ;
                                ; PARTITION NAME
                                ;
000014                          $PNAME::.BLKW   2

                                        .IFF
                                ;
                                ; GCD BLOCK ADDRESS
                                ;
                                GCDA:   .BLKW   1

                                        .ENDC

                                ;
                                ; LOAD SIZE OF PROCESS
                                ;
000020                          $LDSIZ::.BLKW   1


000000                                  .PSECT  ..BUF,D

                                ;
                                ; DISK BUFFER FOR LABEL BLOCK READ
                                ;
```

```
                        .SBTTL  PR.REG - CREATE PROCESS REGION

                ;+
                ;
                ; PR.REG - CREATE PROCESS REGION
                ;
                ; THIS ROUTINE IS CALLED TO CREATE A REGION TO HOLD A NETWORK PROCESS AND
                ; MAP TO THE REGION (TO FORCE THE REGION TO BE LOADED). THE REGION WILL
                ; BE LOCKED IN MEMORY.
                ;
                ; INPUTS:
                ;       R0=LABEL BLOCK BUFFER ADDRESS
                ;       $$PAR=PARTITION NAME (OPTIONAL)
                ;
                ; OUTPUTS:
                ;       C-BIT=SUCCESS/FAILURE
                ;       GCDA=GCD BLOCK ADDRESS
                ;-

PR.REG: MOV     $$PAR+2,$PNAME+2 ; COPY PARTITION OVERRIDE NAME
        MOV     $$PAR,$PNAME     ;   ...
        BNE     10$              ; IF NE, USE OVERRIDE NAME
        MOV     L$BPAR(R0),$PNAME ; ELSE USE PARTITION NAME FROM PROCESS FILE
        MOV     L$BPAR+2(R0),$PNAME+2 ;   ...
10$:    MOV     #^RNT.,$RNAME    ; SET REGION NAME TO 'NT.XXX'
        MOV     $NAME,$RNAME+2   ;   ...
        MOV     BLOX,.PRSIZ      ; SET SIZE OF REGION
        BIT     #FL.TOP,$FLAGS   ; TOP DOWN LOADING REQUIRED?
        BEQ     15$              ; BR IF NO
        BIS     #RS.HLD,R.GSTS+.PRRDB ; SET TOP DOWN LOADING FLAG
15$:    CRRG$S  #.PRRDB          ; CREATE THE PROCESS REGION
        BCS     101$             ; IF CS, CREATE FAILED
        MOV     .PRGID,.WRGID    ; COPY REGION ID FOR MAPPING
        CLR     .WDB+W.NOFF      ;
        CLR     .WDB+W.NLEN
        MAP$S   #.WDB            ; MAP TO THE REGION
        BCS     102$             ; IF CS, MAPPING FAILED
        MOV     R3,-(SP)         ; GET SOME FREE REGISTERS
        MOV     R4,-(SP)         ;   ...
        MOV     .CRTSK,R0        ; GET ADDRESS OF MY ATL NODE
        MOV     A.HA(R0),R0      ; GET ADDRESS OF TASK HEADER
        CALL    $MAPX            ; MAP TO THE TASK HEADER
        MOV     .PRGID,R3        ;; GET THE REGION ID
        DEC     R3               ;; START COUNTING FROM ZERO
        ASH     #2,R3            ;; FORM OFFSET
        ADD     H.ADB(R2),R3     ;; COMPUTE OFFSET INTO HEADER
        BIC     #160000,R3       ;; REMOVE APR BIAS
        BIS     R2,R3            ;;   AND USE CORRECT APR BIAS
        MOV     (R3),R4          ;; GET GCD NODE ADDRESS
        MOV     R4,GCDA          ;; SAVE ADDRESS FOR CLEANUP
        INCB    G.AC(R4)         ;; LOCK REGION IN MEMORY
        INCB    G.IC(R4)         ;;   ...
        MOV     G.PD(R4),R3      ;; GET TPD LISTHEAD FOR REGION
        CALL    ..MUFL           ;; FIND ASSOCIATED MUL ENTRY
        BCS     20$              ;; IF CS, NOT FOUND
        INCB    M.IO(R3)         ;; LOCK TIMESHARING REGION IN MEMORY
20$:    CALL    @(SP)+           ;; C)-ROUTINE RETURN TO RESTORE MAPPING
```

```
                                        .SBTTL  $ALOC2 - ALLOCATE POOL SPACE
                                ;+
                                ; *** - $ALOC2 - ALLOCATE POOL SPACE FROM AREA OUTSIDE TASK'S MAPPING SPACE
                                ;
                                ; THIS ROUTINE WAS ADAPTED FROM THE RSX-11M EXECUTIVE ROUTINE $ALOC1.
                                ; ADDITIONAL CODE WAS ADDED FOR REMAPPING OF THE ADDRESSES IN THE CHAIN.
                                ; FOR ADDITIONAL EXPLANATION OF THIS ROUTINE, CONSULT THE EXECUTIVE
                                ; MODULE "CORAL".
                                ; THIS ROUTINE ALLOCATES POOL SPACE FROM MEMORY THAT IS OUTSIDE THE
                                ; MAPPING SPACE OF THE TASK. THE APR MUST HAVE ALREADY BEEN SET UP BEFORE
                                ; CALLING THIS ROUTINE, WITH THE DISPLACEMENT FOR THE APR PASSED IN R2.
                                ; THIS ROUTINE ASSUMES THAT THE POINTERS IN THE LINKED LIST ARE BIASED BY
                                ; APR6. THIS ROUTINE WILL ENDEAVOR TO PRESERVE THE APR6 MAPPING.
                                ;
                                ; INPUTS:
                                ;       R0 = ADDRESS OF ALLOCATION LISTHEAD-2.
                                ;       R1 = LENGTH OF CORE NEEDED IN BYTES
                                ;       R2 = DISPLACEMENT ADDRESS (100000 OR 120000)
                                ;
                                ; OUTPUTS:
                                ;       C-BIT   SET IF NO MEMORY ALLOCATED
                                ;               CLEAR IF MEMORY WAS ALLOCATED
                                ;
                                ;       IF SUCCESS:
                                ;               R0 = ADDRESS OF THE ALLOCATED BLOCK.
                                ;               R1 = LENGTH OF THE ALLOCATED BLOCK.
                                ;
                                ;-
                                        .IF DF R$$11M
000006                          $ALOC2::
000006  061001                          ADD     (R0),R1         ;ROUND TO NEXT BOUNDARY
000010  042001                          BIC     (R0)+,R1        ;CLEAR EXCESS
000012  000261                          SEC                     ;ASSUME ZERO LENGTH BLOCK
000014  001427                          BEQ     30$             ;IF EQ ZERO LENGTH REQUEST
000016  010246                          MOV     R2,-(SP)        ;SAVE RELOCATION ADDRESS
000020  010002                  10$:    MOV     R0,R2           ;SAVE ADDRESS OF CURRENT BLOCK
000022  011200                          MOV     (R2),R0         ;GET ADDRESS OF NEXT BLOCK
000024  001422                          BEQ     25$             ;IF EQ END OF CHAIN
000026  042700  160000                  BIC     #160000,R0      ;CLEAR CURRENT MAPPING REG.
000032  051600                          BIS     (SP),R0         ;USE OUR MAPPING REG.
000034  026001  000002                  CMP     2(R0),R1        ;BLOCK BIG ENOUGH?
000040  103767                          BLO     10$             ;IF LO NO
000042  001412                          BEQ     20$             ;IF EQ BLOCK IS EXACT SIZE
000044  005720                          TST     (R0)+           ;POINT TO SIZE OF FREE BLOCK
000046  160110                          SUB     R1,(R0)         ;CALCULATE SIZE REMAINING
000050  060001                          ADD     R0,R1           ;POINT TO NEW FREE BLOCK
000052  011011                          MOV     (R0),(R1)       ;COPY SIZE
000054  014041                          MOV     -(R0),-(R1)     ;COPY LINK
000056  010110                          MOV     R1,(R0)         ;STORE LINK TO NEW FREE BLOCK
000060  042710  160000                  BIC     #160000,(R0)    ;ASSUME OTHER PEOPLE USER APR6
000064  052710  140000                  BIS     #140000,(R0)    ;  ...
000070  011012                  20$:    MOV     (R0),(R2)       ;UPDATE LINK TO NEW FREE BLOCK
000072  012602                  25$:    MOV     (SP)+,R2        ;RETORE RELOCATION ADDRESS
000074                          30$:    RETURN
                                        .ENDC
```

```
                                                .SBTTL  REP4A - REPLACE PROCESS NAME
                                        ;+
                                        ; REP4A - REPLACE PROCESS NAME AND CEX DISPATCH TABLE SYMBOL
                                        ;
                                        ; INPUTS:
                                        ;       $NAME=PROCESS NAME
                                        ;       R3=FORMAT STRING ADDRESS
                                        ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESRTOYED
                                        ;       R3,R4=UPDATED
                                        ;-
000066                                  REP4A:  CALL    $REP,P          ; REPLACE PROCESS NAME
000072  112314                          10$:    MOVB    (R3)+,(R4)      ; COPY FORMAT STRING
000074  122724  000052                          CMPB    #'*,(R4)+       ; LOOKING FOR THE SECOND ASTERISK
000100  001374                                  BNE     10$             ;       ...
000102  010400                                  MOV     R4,R0           ; COPY POINTER
000104  112744  000042                          MOVB    #'",-(R4)       ; DEPOSIT A DOUBLE QUOTE
000110  112720  000044                          MOVB    #'$,(R0)+       ; AND "$"
000114  016701  000006                          MOV     $NAME,R1        ; GET PROCESS NAME
000120                                          CALL    $C5TA           ; CONVERT TO ASCII
000124  122740  000040                  20$:    CMPB    #' ,-(R0)       ; SKIP TRAILING BLANKS
000130  001775                                  BEQ     20$             ;       ...
000132  005200                                  INC     R0              ;       ...
000134  112720  000124                          MOVB    #'T,(R0)+       ; DEPOSIT "T"
000140  112720  000102                          MOVB    #'B,(R0)+       ; "B"
000144  112720  000042                          MOVB    #'",(R0)+       ; AND ANOTHER DOUBLE QUOTE
000150  010004                                  MOV     R0,R4           ; SET NEW BUFFER POINTER
000152                                          RETURN
```

```
                                                .SBTTL  $ADJ    - ALLOCATE ADJACENCY DATA BASE
                                        ;+
                                        ; $ADJ  - ALLOCATE ADJACENCY DATA BASE
                                        ;
                                        ; INPUTS:
                                        ;       $PDVA - XPT PDV ADDRESS
                                        ;       R2    - XPT DDB ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;       ADJACENCY DATA BASE ALLOCATED
                                        ;-
000172                                  $ADJ::  CALL    $SAVAL                  ;SAVE REGS
000176  017703  000000G                         MOV     @DECPT,R3               ;GET DECNET HOME BLOCK ADDRESS
000202  005763  000060                          TST     D$NN(R3)                ;ENDNODE ?
000206  001522                                 BEQ     60$                     ;IF YES - BRANCH
000210  016705  000000G                         MOV     $PDVA,R5                ;STORE DPV ADDRESS
000214  005001                                 CLR     R1
000216  156501  000000G                         BISB    Z.LLN(R5),R1            ;GET NUMBER OF CIRCUITS
000222  005201                                 INC     R1                      ;INCLUDE LOOPBACK CHANNEL
000224  066301  000054                          ADD     D$NBRA(R3),R1           ;GET NUMBER OF ROUTER ADJACENCIES
000230  066301  000056                          ADD     D$NBEA(R3),R1           ;INCLUDE NUMBER OF ENDNODE ADJACENCIES
000234  012700  000004                         MOV     #A$LEN,R0               ;SET SIZE OF FIRST ADJACENCY
000240                                          CALL    $MUL                    ;CALCULATE SIZE
000244  000241                                 CLC                             ;CLEAR C-BIT
000246  010104                                 MOV     R1,R4                   ;SAVE TOTAL BYTE SIZE
000250  062701  000077                         ADD     #77,R1                  ;ROUND UP TO NEXT BLOCK
000254  042701  000077                         BIC     #77,R1                  ;  ...
000260                                          ASR$    6,R1                    ;CONVERT TO BLOCKS
000274  010205                                 MOV     R2,R5                   ;SAVE XPT DDB ADDRESS
000276                                          CALL    $ALN18                  ;ALLOCATE BLOCK
000302  103004                                 BCC     10$                     ;  IF SUCCESS - BRANCH
000304                                          EMSG$   9D                      ;TYPE ERROR
000312                                          RETURN

000314  010502                          10$:    MOV     R5,R2                   ;RESTORE XPT DDB ADDRESS
000316  010062  000072                         MOV     R0,N$ADJ1(R2)           ;COPY BIAS OF FIRST ADJACENCY

000322                                  20$:    SWSTK$  40$                     ;ENTER SYSTEM STATE
000326                                          CALL    $MAPX                   ;MAP TO ADJACENCY
000332  005000                                 CLR     R0                      ;  ...
000334  050200                                 BIS     R2,R0                   ;  ...
000336  010502                                 MOV     R5,R2                   ;RESTORE XPT DDB ADDRESS
000340  010401                                 MOV     R4,R1                   ;COPY BYTE COUNT
000342  000241                                 CLC                             ;CLEAR C-BIT
000344  006201                                 ASR     R1                      ;CONVERT SIZE TO WORDS
000346  005762  000074                         TST     N$ADJ2(R2)              ;SECOND ADJACENCY ?
000352  001003                                 BNE     30$                     ;IF YES - BRANCH
000354  005301                                 DEC     R1                      ;COUNT ONE LESS
000356  016320  000014                         MOV     D$LNUM(R3),(R0)+        ;SET UP CHANNEL 0
000362  005020                          30$:    CLR     (R0)+                   ;ZERO OUT BLOCK
000364  005301                                 DEC     R1                      ;COUNT ONE LESS
000366  001375                                 BNE     30$                     ;IF MORE - BRANCH
000370                                          RETURN

000372  005762  000074                  40$:    TST     N$ADJ2(R2)              ;SECOND ADJ ALLOCATED ?
000376  001032                                 BNE     70$                     ;IF YES - BRANCH
000400  010401                                 MOV     R4,R1                   ;STORE TOTAL BYTE SIZE
```

```
                                ;+
                                ; $SYMBL - SEARCH SYMBOL TABLE FILE FOR DEFINITION(S)
                                ;
                                ; THIS ROUTINE IS CALLED TO PROCESS A TASK BUILDER OUTPUT SYMBOL TABLE
                                ; FILE AND TO EXTRACT SYMBOL VALUES.
                                ;
                                ; INPUTS:
                                ;       R0 = FIRST SYMBOL BLOCK ADDRESS
                                ;       R1 = FILE SPEC ADDRESS
                                ;       R2 = FILE SPEC LENGTH
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       R0 = ERROR CODE
                                ;       R1 = NUMBER OF UNDEFINED SYMBOLS
                                ;       R2 = NUMBER OF DEFINED SYMBOLS
                                ;-
000000                          $SYMBL::
000000                                  CALL    $CLSAV          ; SAVE GCL CONTEXT
000004  112767  000000G 000000G        MOVB    #CL.OPN,$CLFLG  ; SET STATUS FOR STB SEARCH
000012                                  CALL    $SAVAL          ; SAVE ALL REGISTERS
000016  016703  000000C                 MOV     $CLFDB+F.DFNB,R3 ; POINT AT GCL'S FDB'S DEFAULT NAMEBLOCK
000022  062703  000000G                 ADD     #N.FTYP,R3      ; AND AT DEFAULT EXTENSION WORD
000026  010346                         MOV     R3,-(SP)        ; SAVE ITS ADDRESS
000030  011346                         MOV     (R3),-(SP)      ; AND ITS CONTENTS
000032  012713  074742                 MOV     #^RSTB,(R3)     ; USE "STB" AS DEFAULT EXTENSION
                                ;;
000036  005066  000006                 CLR     6(SP)           ; ZERO THE ERROR CODE
000042  005066  000010                 CLR     8.(SP)          ; THE SYMBOL COUNT
000046  005066  000012                 CLR     10.(SP)         ; AND THE DEFINITION COUNT
000052  010005                         MOV     R0,R5           ; COPY FIRST SYMBOL BLOCK ADDRESS
000054  001453                         BEQ     10$             ; .. NO BLOCKS AT ALL !
000056  005060  000002          5$:    CLR     SB.DEF(R0)      ; MARK SYMBOL AS UNDEFINED
000062  005060  000010                 CLR     SB.VAL(R0)      ; AND VALUE AS ZERO
000066  005266  000010                 INC     8.(SP)          ; TALLY THE BLOCK
000072  011000                         MOV     (R0),R0         ; GET NEXT BLOCK IN LIST
000074  001370                         BNE     5$              ; .. NOT AT END YET
                                ;;
000076                                  CALL    $CLOUE          ; OPEN STB FILE
000102  105777  000000G                 TSTB    @$CLIOS         ; SUCCESS ?
000106  100441                         BMI     111$            ; NO ..
                                ;;
000110                          10$:    CALL    READ            ; READ NEXT RECORD
000114  103424                         BCS     50$             ; .. READ FAILURE
000116                                  CALL    CHECK           ; CHECK STB RECORD FORMAT
000122  103417                         BCS     40$             ; .. BAD FORMAT
000124  102771                         BVS     10$             ; .. NOT GSD RECORD
000126  010500                         MOV     R5,R0           ; GET FIRST SYMBOL BLOCK ADDRESS
000130  001414                         BEQ     40$             ; .. END GSD RECORD SEEN
000132  005760  000002          20$:   TST     SB.DEF(R0)      ; IS THIS SYMBOL ALREADY DEFINED ?
000136  001006                         BNE     30$             ; YES ..
000140                                  CALL    LOOK            ; LOOK FOR IT WITHIN CURRENT RECORD
000144  026666  000010  000012         CMP     8.(SP),10.(SP)  ; ALL SYMBOLS DEFINED ?
000152  001403                         BEQ     40$             ; YES ..
000154  011000                 30$:    MOV     (R0),R0         ; GET NEXT SYMBOL BLOCK ADDRESS
000156  001365                         BNE     20$             ; .. KEEP CHECKING THIS RECORD
000160  000753                         BR      10$             ; END OF LIST .. GET NEXT RECORD
```

```
                                ;+
                                ; $CNB - ".CNB" ACTION ROUTINE
                                ; $CNW - ".CNW" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $SLTA=SLT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;-
000126                          $CNB::
000126                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000132  016700  000000G                 MOV     $SLTA,R0        ; GET SLT ADDRESS
000136  116067  000012  000000G         MOVB    L.CTL(R0),$VALUE ; COPY CONTROLLER NUMBER INTO $VALUE
000144                                  CALLR   $BYTE           ; PRETEND IT'S A ".BYTE"
                                ;
000150                          $CNW::
000150                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000154  016700  000000G                 MOV     $SLTA,R0        ; GET SLT ADDRESS
000160  005001                          CLR     R1              ; GET CONTROLLER NUMBER AS A WORD VALUE
000162  156001  000012                  BISB    L.CTL(R0),R1    ;
000166  010167  000000G                 MOV     R1,$VALUE       ; COPY IT INTO $VALUE
000172                                  CALLR   $WORD           ; PRETEND IT'S A ".WORD"
```

```
                                        ;+
                                        ; $WORD - ".WORD" ACTION ROUTINE
                                        ; $ZWORD - ACTION ROUTINE TO STORE A ZERO WORD
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=WORD VALUE
                                        ;
                                        ; OUTPUTS:
                                        ;       R2=DESTROYED
                                        ;-
001026                                  $ZWORD::
001026  005067  000000G                         CLR     $VALUE          ; USE ZERO AS EXPRESSION VALUE
001032                                  $WORD::
001032                                          CALL    $RQ3            ; REQUEST BINARY FILE
001036  103411                                  BCS     10$             ; IF CS, NO ACCESS
001040  112722  000000G                         MOVB    #.WORD,(R2)+    ; STORE FUNCTION CODE
001044  116722  000000G                         MOVB    $VALUE,(R2)+    ; STORE WORD VALUE
001050  116712  000001G                         MOVB    $VALUE+1,(R2)
001054  062767  000002  000000G                 ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
001062                                  10$:    RETURN
```

B 1
C 1
D 1
E 1
F 1
G 1
H 1
I 1
J 1
K 1
L 1
M 1
N 1
B 2
C 2
D 2
E 2
F 2
G 2
H 2
I 2
J 2
K 2
L 2
M 2
N 2
B 3
C 3
D 3
E 3
F 3
G 3
H 3
I 3
J 3
K 3
L 3
M 3
N 3
B 4
C 4
D 4
E 4
F 4
G 4
H 4
I 4
J 4
K 4
L 4
M 4
N 4
B 5
C 5
D 5
E 5
F 5
G 5
H 5
I 5

J 5
K 5
L 5
M 5
N 5
B 6
C 6
D 6
E 6
F 6
G 6
H 6
I 6
J 6
K 6
L 6
M 6
N 6
B 7
C 7
D 7
E 7
F 7
G 7
H 7
I 7
J 7
K 7
L 7
M 7
N 7
B 8
C 8
D 8
E 8
F 8
G 8
H 8
I 8
J 8
K 8
L 8
M 8
N 8
B 9
C 9
D 9
E 9
F 9
G 9
H 9
I 9
J 9
K 9
L 9
M 9
N 9
B 10
C 10
D 10

NTL11M

POLAL

PRALL

PRIO

E 10
F 10
G 10
H 10
I 10
J 10
K 10
L 10
M 10
N 10
B 11
C 11
D 11
E 11
F 11
G 11
H 11
I 11
J 11
K 11
L 11
M 11
N 11
B 12
C 12
D 12
E 12
F 12
G 12
H 12
I 12
J 12
K 12
L 12
M 12
N 12
B 13
C 13
D 13
E 13
F 13
G 13
H 13
I 13
J 13
K 13
L 13
M 13
N 13
B 14
C 14
D 14
E 14
F 14
G 14
H 14
I 14
J 14
K 14
L 14

PROAL

PROERR

PROXPT

SYMBL

TMPACT

M 14
N 14
B 15
C 15
D 15
E 15
F 15
G 15
H 15
I 15
J 15
K 15
L 15
M 15
N 15
B 16
C 16
D 16
E 16
F 16
G 16
H 16
I 16
J 16
K 16
L 16
M 16

TMPACT CREATED BY MACRO ON 29-JUN-85 AT 00:38 PAGE 5

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	6-125	6-129	7-151	8-169	8-175	9-192	9-193	10-210	10-211	11-227
	11-228	12-251	13-265	14-299	14-302	14-304	15-322	16-340	17-359	17-360
	18-393	18-394	19-433	19-434	20-458	20-459	20-480	20-485	21-506	22-526
	22-532	23-576	24-603	27-664						
CALLR	8-172	8-180								
EMSG$	#5-59	12-251	23-576	24-603	27-664					
EMSG$R	#5-59	13-277	13-278	26-649	26-650					
NTLER$	#5-59	5-74	5-75	5-76	5-77	5-78	5-79	5-80	5-81	
RETURN	6-134	7-156	9-198	10-215	11-232	12-252	13-272	14-308	15-326	16-345
	17-364	18-398	19-438	20-463	20-489	21-512	22-530	22-541	22-544	23-578
	24-605	25-618	26-644	27-665						
SLTDF$	#5-59	5-61								
SOB	26-641									

FILEID**TMPACO

```
                                ;
                                ; ERROR CONDITIONS
                                ;
000252                  111$:   EMSG$R  23              ; INTERRUPT LINKAGE ERROR
000260                  121$:   EMSG$R  24              ; VECTOR NOT IN SYSTEM
000266                  131$:   EMSG$R  25              ; VECTOR NOT AVAILABLE
```

```
                                40$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000610                          101$:   EMSG$R  41              ; ".MPLHD" OPERATOR MISSING
000616                          111$:   EMSG$R  42              ; ".LFLHD" OPERATOR MISSING
000624                          CTLERR: EMSG$R  22              ; FILE COUNT TOO LARGE

                                        .DSABL  LSB
```

```
                                        ;+
                                        ; REP2A - REPLACE PROCESS TYPE AND SYMBOL NAME
                                        ;
                                        ; INPUTS:
                                        ;       R3=FORMAT STRING ADDRESS
                                        ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                        ;       ASYM=ASCII SYMBOL (6 CHARACTERS)
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1=DESTROYED
                                        ;       R3,R4=UPDATED
                                        ;-
000210                                  REP2A:  CALL    $REP.C          ; REPLACE PROCESS NAME
000214  112314                          10$:    MOVB    (R3)+,(R4)      ; LOOK FOR SECOND ASTERISK
000216  122724  000052                          CMPB    #'*,(R4)+       ;       ...
000222  001374                                  BNE     10$             ;       ...
000224  112764  000042  177777                  MOVB    #'",-1(R4)      ; DEPOSIT DOUBLE QUOTE
000232  012700  000000'                        MOV     #ASYM,R0        ; GET ASCII SYMBOL ADDRESS
000236  012701  000006                        MOV     #6,R1           ; MAXIMUM OF 6 CHARACTERS
000242  112014                          20$:    MOVB    (R0)+,(R4)      ; LOOK FOR FIRST BLANK
000244  122724  000040                          CMPB    #' ,(R4)+       ;       ...
000250  001403                                  BEQ     30$             ;       ...
000252                                          SOB     R1,20$          ; OR 6 NON-BLANKS
000256  005204                                  INC     R4              ; NO BLANK TO CLOBBER
000260  112764  000042  177777          30$:    MOVB    #'",-1(R4)      ; STORE ANOTHER DOUBLE QUOTE
000266                                          RETURN

000001                                          .END
```

```
                                ;+
                                ; MAX14 - CHECK FOR NO MORE THAN 14. ALLOCATIONS
                                ;
                                ; INPUTS:
                                ;       $NALL=NUMBER OF ALLOCATIONS
                                ;       $NLIB=NUMBER OF LIBRARIES
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0=DESTROYED
                                ;-
000574  116700  000000G         MAX14:  MOVB    $NALL,R0        ; GET NUMBER OF ALLOCATIONS
000600  010046                          MOV     R0,-(SP)        ; SAVE IT
000602  116700  000000G                 MOVB    $NLIB,R0        ; GET NUMBER OF LIBRARIES
000606  062600                          ADD     (SP)+,R0        ; ADD THE TWO TOGETHER
000610  022700  000015                  CMP     #13.,R0         ; ARE WE AT THE MAXIMUM ALREADY?
000614  103401                          BCS     101$            ; IF CS, NO MORE ALLOWED
000616                                  RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000620                          101$:   EMSG$R  78              ; ONLY 14. ALLOCATIONS/LIBRARIES ALLOWED
```

```
    244 000310  116777  0000000G 0000000G 101$:   MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
    245 000316                                   EMSG$R  72              ; LABEL BLOCK READ FAILURE
    246 000324                             111$:  EMSG$R  73              ; INVALID BINARY IMAGE
    247 000332                             121$:  EMSG$R  74              ; SIZE EXCEEDS 32K-32 WORDS
    248 000340                             131$:  EMSG$R  75              ; EXTENSION EXCEEDS 32K WORDS
    249
    250
    251         000001                           .END
```


Symbol table

```
                                        .SBTTL  $LIBR1 - .LIBR OPERAND 1
                                ;+
                                ; $LIBR1 - ".LIBR" OPERAND 1 ACTION ROUTINE
                                ;
                                ; THIS ROUTINE IS CALLED TO LOOK FOR AN EXISTING LIBRARY WHOSE NAME MATCHES
                                ; THE FILE NAME FROM THE TEMPLATE LINE. IF NO MATCH IS FOUND, THE TPARS
                                ; TRANSITION IS REJECTED.
                                ;
                                ; INPUTS:
                                ;       .PSTPT=TPARS STRING POINTER
                                ;       .PSTCN=TPARS COUNT
                                ;
                                ; OUTPUTS:
                                ;       F2.LBR SET IF A MATCH IS FOUND
                                ;       $LBIAS=LIBRARY DESCRIPTOR BIAS (MAPPED ONLY)
                                ;       $LBADR=LIBRARY DESCRIPTOR ADDRESS OR 0
                                ;-
000106                          $LIBR1::
000106  032767  000000G 000000G         BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN?
000114  001517                          BEQ     65$             ; IF EQ, NO
000116  016767  000000G 000042'         MOV     $FLAGS,.FLAGS   ; MAKE A LOCAL COPY OF $FLAGS
000124  016700  000000G                 MOV     .PSTPT,R0       ; GET TPARS POINTER
000130  012701  000030'                 MOV     #ASCII,R1       ; LOCAL DATA ADDRESS
000134  016702  000000G                 MOV     .PSTCN,R2       ; AND TPARS LENGTH
                                
                                        .IF DF  R$$MPL
                                        SAVRG   <R3,R2>         ; SAVE REGS
                                        MOV     #$LIBNM,R3      ; GET BUFFER ADDRESS
                                        MOV     #10.,R2         ; GET BUFFER SIZE
                                1$:     CLRB    (R3)+           ; SPACE OUT BUFFER
                                        SOB     R2,1$           ;       ...
                                        RESRG   <R2>            ; RESTORE .PSTCN
                                        MOV     #$LIBNM,R3      ; GET BUFFER ADDRESS
                                
                                5$:     MOVB    (R0),(R3)+      ; COPY FILE NAME
                                        .IFTF
                                
000140  112021                  10$:    MOVB    (R0)+,(R1)+     ; COPY FILE NAME
                                
                                        .IFF
000142                                  SOB     R2,10$          ;       ...
                                        .IFT
                                        SOB     R2,5$           ;       ...
                                        RESRG   <R3>            ; RESTORE REG
                                        .ENDC
                                
000146  012702  000011                  MOV     #9.,R2          ; BLANK FILL OUT TO 9 CHARACTERS
000152  166702  000000G                 SUB     .PSTCN,R2       ;
000156  001404                          BEQ     30$             ; IF EQ, NAME WAS 9 LONG
000160  112721  000040          20$:    MOVB    #' ,(R1)+       ;       ...
000164                                  SOB     R2,20$          ;       ...
000170  012700  000030'          30$:    MOV     #ASCII,R0       ; CONVERT FILE NAME TO RAD50
000174                                  CALL    $CAT5           ;       ...
000200  010167  000000G                 MOV     R1,$FNR50       ;       ...
000204                                  CALL    $CAT5           ;       ...
000210  010167  000002G                 MOV     R1,$FNR50+2     ;       ...
000214                                  CALL    $CAT5           ;       ...
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
X2BHI	= 000003	#6-73 16-475
X2BLK	= 000002	#6-74 15-431 16-477
X2BLO	= 000001	#6-72 16-473
X2WHI	= 000005	#6-71 15-435
X2WLO	= 000004	#6-70 15-433
X3BHI	= 000013	#6-69 16-485
X3BLO	= 000003	#6-68 16-483
X3WHI	= 000002	#6-67 15-443
X3WLO	= 000001	#6-66 15-441
$AZFS	= ****** GX	10-297
$BFPTR	= ****** GX	*10-286
$CAPR	= ****** GX	*9-168 *10-284
$CAT5	= ****** GX	10-228 10-230 10-232
$CORE2	000000 RG	#9-152
$ELINE	= ****** GX	8-117 8-118 8-120 8-121 8-122
$ERRZK	000106 R	#8-118 13-391
$ERR39	000044 R	#8-117 9-163
$ERR89	000150 R	#8-120 17-511
$ERR90	000176 R	#8-121 18-529
$ERR91	000232 R	#8-122 15-445 16-487
$FILEN	= ****** GX	10-295
$FLAGS	= ****** GX	10-196 17-509 18-527
$FLAG1	= ****** GX	9-153 11-329 *13-377 *14-409
$FLAG2	= ****** GX	9-169 10-194 10-276 *10-288 *11-336 11-339
$FNR50	= ****** GX	*10-229 *10-231 *10-233 11-320 11-322 11-324
$LBADR	= ****** GX	*10-234 *11-337
$LBIAS	= ****** GX	*11-314
$LIBR1	000106 RG	#10-193
$LINKS	000574 RG	#12-354
$MAPX	= ****** GX	10-238 13-384
$REP.C	= ****** GX	8-117 8-118 8-120 8-121 8-122
$RQ1	= ****** GX	12-356 15-449 16-491
$RQ7	= ****** GX	10-285
$SIZE	= ****** GX	*12-359 *15-451 *16-493
$TERR	= ****** GX	8-118 8-120 8-121 8-122
$TWARN	= ****** GX	8-117
$UMROF	000700 RG	#14-408
$UMRON	000622 RG	#13-376
$VALUE	= ****** GX	9-155 *9-166 *10-283 15-431 15-433 15-435 15-441 15-443 15-447 16-473 16-475 16-477 16-483 16-485 16-489
$VFLLC	001156 R	12-355 15-446 16-488 #17-508
$VFLV1	= ****** GX	10-280
$VXLLC	001176 R	15-437 16-479 #18-527
$X2CHB	001034 RG	#16-472
$X2CHW	000710 RG	#15-430
$X3CHB	001076 RG	#16-482
$X3CHW	000752 RG	#15-440
.BASEB	= 140000	#5-62 7-98 10-283
.CXLBR	= ****** GX	10-247
.CXLB1	= ****** GX	10-241
.FLAGS	000042 R	#7-108 *10-196 11-327
.LINKS	= ****** GX	12-358

```
                                        ;+
                                        ; REP6 - REPLACE LIBRARY NAME AND THEN FORM APR NUMBER
                                        ;
                                        ; INPUTS:
                                        ;       R3=FORMAT STRING ADDRESS
                                        ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                        ;       $CAPR=APR NUMBER IN BINARY
                                        ;
                                        ; OUTPUTS:
                                        ;       R0=DESTROYED
                                        ;       R3,R4=UPDATED
                                        ;-
                                                .IF DF  M$$MGE

000574  112314                 REP6:   MOVB    (R3)+,(R4)      ; COPY FORMAT STRING
000576  122724  000052                  CMPB    #'*,(R4)+       ; LOOKING FOR FIRST ASTERISK
000602  005304                          DEC     R4              ; SKIP THE ASTERISK
000604  012700  000000G                 MOV     #$FILEN,R0      ; GET ASCIZ FILE NAME ADDRESS
000610  112024                 10$:    MOVB    (R0)+,(R4)+     ; COPY THE FILE NAME
000612  001376                          BNE     10$             ;       ...
000614  005304                          DEC     R4              ; SKIP THE ZERO BYTE
000616  010500                          MOV     R5,R0           ; COPY ERROR MESSAGE BLOCK ADDRESS
000620  005720                          TST     (R0)+           ; NOW POINTS AT TEXT STRING
000622  122720  000120         20$:    CMPB    #'P,(R0)+       ; LOOK FOR "P"
000626  001375                          BNE     20$             ;       ...
000630  122720  000122                  CMPB    #'R,(R0)+       ; FOLLOWED BY "R"
000634  001372                          BNE     20$             ;       ...
000636  112710  000060                  MOVB    #'0,(R0)        ; FORM APR NUMBER
000642  156710  000000G                 BISB    $CAPR,(R0)      ;       ...
000646                                  RETURN

                                        .ENDC


000001                                  .END
```

```
                        .TITLE  TMPAC9 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 9
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
```

```
                                        .SBTTL  $TMALT - ALLOCATE LINE TABLE
                                ;+
                                ; $TMALT - ALLOCATE LINE TABLE FROM DYNAMIC STORAGE
                                ;
                                ; THIS ROUTINE IS CALLED TO ALLOCATE THE PROCESS LINE TABLE FROM DYNAMIC
                                ; STORAGE AND TO SAVE ITS ADDRESS FOR THE BINARY EXPANSION ROUTINES.
                                ;
                                ; INPUTS:
                                ;       $SIZE=NUMBER OF BYTES NEEDED
                                ;
                                ; OUTPUTS:
                                ;       R0,R1   = DESTROYED
                                ;       $ADDR   = LINE TABLE ADDRESS
                                ;       $ICBAD  = ADDRESS OF ICB (ONLY FOR DDM'S)
                                ;       $ICBSZ  = SIZE OF ICB
                                ;                 NOTE: ALL ICB'S FOR A CONTROLLER ARE ALLOCATED CONTIGUOUSLY.
                                ;-
                                        .ENABL  LSB
000000  005767  000000G        $TMALT::TST     $FLAG1                  ;ANY ERROR YET?
000004  100507                         BMI     30$                     ;BR IF YES, DO NOTHING
000006  032767  000000G 000000G        BIT     #FL.CHA,$FLAGS          ; MUX CHARACTERISTICS UPDATE ONLY ?
000014  001103                         BNE     30$                     ; YES ...
000016  032767  000000G 000000G        BIT     #FL.DDM,$FLAGS          ; IS THIS A DDM ?
000024  001411                         BEQ     5$                      ; IF EQ, NO
000026  016701  000000G                MOV     $PDVA,R1                ; GET PDV ADDRESS
000032  032761  000000G 000000G        BIT     #ZF.PSE,Z.FLG(R1)       ; IS THIS A PSEUDO-DEVICE?
000040  001003                         BNE     5$                      ; BR IF YES
000042                                 CALL    KRBAL                   ; ALLOCATE THE KRB
000046  103466                         BCS     30$                     ; IF CS, FAILURE
000050  016701  000000G        5$:     MOV     $SIZE,R1                ;GET BYTE COUNT NEEDED
000054  001464                         BEQ     101$                    ;IF ZERO, ERROR

000056  062701  000003         10$:    ADD     #3,R1                   ;ROUND UP NEXT MULTIPLE OF 4
000062  042701  000003                 BIC     #3,R1                   ;
000066  010167  000000G                MOV     R1,$SIZE                ;RESET LINE TABLE SIZE

                                        .IF DF  R$$MPL
                                        MOV     FMASK,FM2TMP            ;GET $FMASK ADDRESS
                                        ADD     #2,FM2TMP               ;$FMASK+2
                                        BIT     #F2.DAS,@FM2TMP         ;IS DATA SPACE ENABLED ?
                                        BEQ     11$                     ; IF EQ, NO - JUST LIKE 11M
                                        BIT     #FL.DDM,$FLAGS          ;IS THIS A DDM?
                                        BEQ     11$                     ;IF EQ, NO
                                        CALL    $ALP16                  ;TRY TO ALLOCATE SPACE FROM PROCESS
                                        BR      12$                     ;AND CONTINUE
                                11$:    CALL    $ALPRO                  ;TRY TO ALLOCATE FROM PRO/CEX/RSX
                                12$:
                                        .IFF
000072                                  CALL    $ALPRO                  ;ALLOCATE SINGLEWORD STORAGE
                                        .ENDC
000076  103465                         BCS     111$                    ;BR IF FAILURE
000100  010067  000000G                MOV     R0,$ADDR                ;SET LINE TABLE ADDRESS
```

```
                                        SUB     #2,R3                   ;* ...
                                        CALL    DEAC2                   ;* DE-ALLOCATE THE ICB
                                        MOV     (SP)+,R0                ;* IS THERE ANOTHER ICB ?
                                        BEQ     30$                     ;* IF EQ, NO
                                        MOV     R2,-(SP)                ;* DOES POINTER CONTAIN A CPU NUMBER ?
                                        BNE     10$                     ;* IF NE, YES
                                        CRASH                           ;*** ONLY 1 ICB PER CONTROLLER  ??? ***;
                                30$:    MOV     (SP)+,@KSAR5            ;* RESTORE PREVIOUS MAPPING
                                        BR      50$                     ;* AND LEAVE

                                        .IFTF   ; DF R$$MPL

001352  016700  000000G        40$:    MOV     $ICBAD,R0               ;* GET THE ICB ADDRESS
001356  001404                          BEQ     50$                     ;* IF EQ, NONE - SO LEAVE
001360  016701  000000G                 MOV     $ICBSZ,R1               ;* GET THE LENGTH OF THE ICB

                                        .IFT    ; DF R$$MPL

                                        TST     -(R0)                   ;* INCLUDE COUNT AND SIZE FIELDS
                                        MOV     ICAVL,R2                ;* GET ADDRESS OF ICB LISTHEAD
                                        SUB     #2,R2                   ;* ...
                                        CALL    @DEAC1                  ;* DE-ALLOCATE THE ICB BLOCK

                                        .IFF    ; DF R$$MPL

001364                                  CALL    @DEACB                  ;* DE-ALLOCATE THE ICB BLOCK

                                        .ENDC   ; DF R$$MPL

001370                          50$:    RETURN                          ;* BACK TO USER MODE AND TO CALLER
                                        .ENDC   ; DF    R$$11M & M$$MGE
```

TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 3

SYMBOL CROSS REFERENCE CREF 04 00

SYMBOL	VALUE	REFERENCES							
UBLTA	001204 R	11-549	11-597	#13-674					
XICB	001312 R	8-301	11-636	#14-710					
XLT	001272 R	8-302	11-637	#14-701					
ZF.PSE	= ****** GX	8-175	8-212	10-449	14-716				
Z.AVL	= ****** GX	15-843							
Z.FLG	= ****** GX	8-175	8-212	10-449	14-716				
$ADDR	= ****** GX	*8-205	*8-296	13-678	14-701	*14-703	14-718		
$ALN18	= ****** GX	11-581							
$ALPRO	= ****** GX	8-200							
$DEAC2	= ****** GX	15-847							
$DEA16	= ****** GX	15-879							
$ERR33	000012 R	#7-103	8-299						
$ERR34	000044 R	#7-104	8-300						
$ERR35	000112 R	#7-105	10-473						
$ERR36	000150 R	#7-106	10-474						
$ERR37	000216 R	#7-107	10-475						
$ERR84	000260 R	#7-108	10-476						
$ERR86	000322 R	#7-109	11-638						
$ERR9B	000372 R	#7-110	8-303						
$ERR9C	000432 R	#7-111	9-428						
$ERR9Y	000472 R	#7-112							
$FLAGS	= ****** GX	8-170	8-172	8-209	8-297	10-446	14-713		
$FLAG1	= ****** GX	8-168	10-444						
$ICBAD	= ****** GX	*8-275	14-765						
$ICBLN	= ****** GX	8-241							
$ICBSZ	= ****** GX	*8-214	*8-244	14-767					
$KRBLN	= ****** GX	9-330							
$LDA	= ****** GX	*11-508	11-546	*13-693					
$MAPX	= ****** GX	11-515	11-590	15-839					
$NALL	= ****** GX	11-499	13-682	13-686	13-689				
$NLIB	= ****** GX	11-500							
$NVECT	= ****** GX	8-240	10-451						
$PCB	= ****** GX	15-829							
$PDVA	= ****** GX	8-174	8-211	10-448	12-657	14-715	15-816		
$PRAVL	= ****** GX	15-842	15-848						
$REP.P	= ****** GX	7-103	7-104	7-105	7-106	7-107	7-108	7-109	7-110	7-111
$SIZE	= ****** GX	8-179	*8-184	13-679	14-702				
$SLTA	= ****** GX	9-371							
$TERR	= ****** GX	7-104	7-105	7-106	7-107	7-108	7-109	7-110	7-111	7-112
$TMALT	000000 RG	#8-168							
$TMCHK	000450 RG	#10-443							
$TWE	= ****** GX	7-103							
$UBIAS	= ****** GX	*13-675							
$UDA	= ****** GX	*11-498	11-575	*13-688					
$$CSR	= ****** GX	9-367							
$$PRI	= ****** GX	9-362							
$$VECT	= ****** GX	9-363							
.CXUNL	= ****** GX	11-516	11-599	*11-602					
.DEA16	001372 R	14-704	#15-800						

TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 4

```
TMPEXP - NTL TEMPLATE EXPANSION MACRO V05.03b Saturday 29-Jun-85 00:42  Page 5-2
Symbol table

A$$CHK= 000000          F$$LVL= 000001          L$$P11= 000001          N$$MOV= 000010          T$$MIN= 000000
A$$CPS= 000000          G$$TFP= 000000          L$$11R= 000000          N$$NCT= 000001          V$$CTR= 001000
A$$PRI= 000000          G$$TSS= 000000          M$$CRB= 000124          N$$PEM= 000001          X$$DBT= 000000
A$$TRP= 000000          G$$TTK= 000000          M$$CRX= 000000          P$$P45= 000000          $ADDR = ****** GX
C$$CKP= 000000          G$$WRD= 000000          M$$FCS= 000000          P$$WRD= 000000          $FLAGS= ****** GX
C$$ORE= 000400          I$$RAR= 000000          M$$MGE= 000000          Q$$OPT= 000010          $FLAG1= ****** GX
C$$RSH= 177564          I$$RDN= 000000          M$$NET= 000000          R$$DER= 000000          $NALL = ****** GX
D$$BUG= 177514          K$$CNT= 177546          M$$OVR= 000000          R$$K11= 000001          $TMEXP  000000RG
D$$ISK= 000000          K$$CSR= 177546          N$$ACC= 000001          R$$SND= 000000          $TMEX2= ****** GX
D$$L11= 000001          K$$LDC= 000000          N$$BUF= 000001          R$$11M= 000000          $TMEX3= ****** GX
D$$YNC= 000000          K$$TPS= 000074          N$$LDV= 000001          S$$WRG= 000000          $TMEX5= ****** GX
D$$YNM= 000000          LD$LP = 000000          N$$MCP= 000001          S$$YSZ= 007600          $TMEX6= ****** GX
E$$XPR= 000000          L$$ASG= 000000          N$$MLL= 000001          T$$KMG= 000000          $TMEX7= ****** GX
FL.CHA= ****** GX       L$$DRV= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000112    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 8920 Words  ( 35 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:05.38
SY:TMPEXP.V2,[132,134]TMPEXP/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPEXP
```



```
TMPEXP     CREATED BY  MACRO  ON 29-JUN-85 AT 00:42     PAGE 1
```

```
                                        ;
                                        ; SYSTEM LINE NUMBER (WORD)
                                        ;
000470  005000                  SLNW:   CLR     R0              ; ASSUME NORMAL EXPANSION
000472  005767  000154'                 TST     FLAGS           ; IS IT MUX EXPANSION?
000476  100003                          BPL     10$             ; IF PL, NO
000500  016700  000152'                 MOV     INDEX,R0        ; GET MUX INDEX
000504  122020                          CMPB    (R0)+,(R0)+     ; SKIP DLC LINE NUMBER
000506  005003                  10$:    CLR     R3              ; GET LINE NUMBER AS A WORD
000510  156003  000000G                 BISB    $SLN(R0),R3     ;
000514  010324                          MOV     R3,(R4)+        ; STORE IT
000516                                  RETURN
                                        ;
                                        ; LOCAL NODE ADDRESS
                                        ;
000520  017700  000000G         NOD:    MOV     @DECPT,R0       ; STORE DECNET HOME BLOCK POINTER
000524  016024  000014                  MOV     D$LNUM(R0),(R4)+ ; COPY LOCAL NODE ADDRESS
000530                                  RETURN
                                        ;
                                        ; PROTOCOL OVERHEAD SIZE
                                        ;
000532  112524                  PRTOV:  MOVB    (R5)+,(R4)+     ; COPY PROTOCOL OVERHEAD SIZE
000534                                  RETURN
                                        ;
                                        ; STATION NUMBER (BYTE)
                                        ;
000536  105767  000154'         STNB:   TSTB    FLAGS           ; IS IT MTP EXPANSION?
000542  100404                          BMI     10$             ; IF MI, YES
000544  022767  000001  000000G         CMP     #1,$$MTP        ; ARE WE DEALING WITH ONLY 1 STATION ?
000552  001005                          BNE     101$            ; IF NE, NO
000554  016700  000152'         10$:    MOV     INDEX,R0        ; GET CURRENT STATION INDEX
000560  116024  000000G                 MOVB    $$SNUM(R0),(R4)+ ; STORE STATION NUMBER
000564                                  RETURN
000566  112724  177777          101$:   MOVB    #-1,(R4)+       ; DLC STNB NOT IN MPTAB FILE
000572                                  RETURN
                                        ;
                                        ; STATION NUMBER (WORD)
                                        ;
000574  105767  000154'         STNW:   TSTB    FLAGS           ; IS IT MTP EXPANSION?
000600  100404                          BMI     10$             ; IF MI, YES
000602  022767  000001  000000G         CMP     #1,$$MTP        ; ARE WE DEALING WITH ONLY 1 STATION ?
000610  001007                          BNE     101$            ; IF NE, NO
000612  016700  000152'         10$:    MOV     INDEX,R0        ; GET CURRENT STATION INDEX
000616  005003                          CLR     R3              ; GET STATION NUMBER
000620  156003  000000G                 BISB    $$SNUM(R0),R3   ;
000624  010324                          MOV     R3,(R4)+        ; STORE IT
000626                                  RETURN
000630  012724  177777          101$:   MOV     #-1,(R4)+       ; DLC STNW NOT IN MPTAB FILE
000634                                  RETURN

                                        ;
```

TMPEX2 - NIL EXPANSION ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:42 Page 17-2
Symbol table

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 12325 Words (49 Pages)
Size of core pool: 14440 Words (55 Pages)
Operating system: RSX-11M/PLUS

Elapsed time: 00:00:20.31
SY:TMPEX2.V2,[132,134]TMPEX2/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPEX2

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 1

```
**FILE**ID**TMPACO
```

```
TTTTTTTTTT    MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC    000000
TTTTTTTTTT    MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC    000000
    TT        MMMM  MMMM   PP      PP    AA      AA   CC          00    00
    TT        MMMM  MMMM   PP      PP    AA      AA   CC          00    00
    TT        MM  MM  MM   PP      PP    AA      AA   CC          00  0000
    TT        MM  MM  MM   PP      PP    AA      AA   CC          00  0000
    TT        MM      MM   PPPPPPPP      AA      AA   CC          00  00 00
    TT        MM      MM   PPPPPPPP      AA      AA   CC          00  00 00
    TT        MM      MM   PP            AAAAAAAAAA   CC          0000   00
    TT        MM      MM   PP            AAAAAAAAAA   CC          0000   00
    TT        MM      MM   PP            AA      AA   CC          00     00      ....
    TT        MM      MM   PP            AA      AA   CC          00     00      ....
    TT        MM      MM   PP            AA      AA     CCCCCCCC    000000      ....
    TT        MM      MM   PP            AA      AA     CCCCCCCC    000000      ....
```

```
LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```



```
TMPACO - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:38
Table of contents
```

```
                                        ;+
                                        ; REP2B - REPLACE PROCESS TYPE AND VECTOR VALUE
                                        ;
                                        ; INPUTS:
                                        ;       R2=VECTOR VALUE
                                        ;       R3=FORMAT STRING ADDRESS
                                        ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       R3,R4=UPDATED
                                        ;-
000274                                  REP2B:  CALL    $REP.C          ; REPLACE PROCESS NAME
000300  112314                         10$:    MOVB    (R3)+,(R4)      ; LOOK FOR SECOND ASTERISK
000302  122724  000052                          CMPB    #'*,(R4)+       ;       ...
000306  001374                                  BNE     10$             ;       ...
000310  005304                                  DEC     R4              ; SKIP THE ASTERISK
000312  010400                                  MOV     R4,R0           ; COPY BUFFER ADDRESS
000314  010201                                  MOV     R2,R1           ; COPY VECTOR VALUE
000316  005002                                  CLR     R2              ; SUPPRESS LEADING ZEROES
000320                                          CALL    $CBOMG          ; CONVERT TO OCTAL
000324  010004                                  MOV     R0,R4           ; SET NEW BUFFER ADDRESS
000326                                          RETURN
                                        ;
000001                                          .END
```

```
                                ;+
                                ; $MXTAB - ".MXTAB" ACTION ROUTINE
                                ; $CFILE - ".FILE" ACTION ROUTINE IF COUNT IS SPECIFIED
                                ;
                                ; INPUTS:
                                ;       $$MUX/$VALUE=FILE REPEAT COUNT
                                ;       $FILEN=SECONDARY TEMPLATE FILE NAME
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;-

                                        .ENABL  LSB

000632                          $MXTAB::
000632                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000636                                  CALL    $VFMUX          ; AND A MUX LINE
000642  012700  000000G                 MOV     #F1.MXT,R0      ; MAKE SURE THIS IS THE ONLY OCCURANCE
000646                                  CALL    $VFSPC          ;       ...
000652  016767  000000G 000000G         MOV     $SIZE,$OFF      ; USE CURRENT SIZE AS OFFSET VALUE
000660  016767  000000G 000000G         MOV     $$MUX,$VALUE    ; DO IT <UNIT COUNT> TIMES
000666  012700  000000G                 MOV     #.MXTAB,R0      ; FUNCTION CODE TO BE USED
000672                                  CALL    5$              ; PROCESS SECONDARY TEMPLATE
000676  010067  000000G                 MOV     R0,$MXLEN       ; SAVE MUX TABLE LENGTH
000702                                  RETURN
000704                          $CFILE::
000704                                  CALL    $EVEN           ; AUTOMATIC WORD ALIGNMENT ($FILE NOT ALIGNED)
000710  012700  000000G                 MOV     #.FILE,R0       ; FUNCTION CODE TO BE USED
000714  022767  000400  000000G 5$:     CMP     #256.,$VALUE    ; IS COUNT > 256.?
000722  103740                          BLO     CTLERR          ; IF LO, YES
000724  005767  000000G                 TST     $VALUE          ; IS COUNT 0?
000730  001432                          BEQ     30$             ; IF EQ, YES
000732                                  CALL    $RQ2            ; REQUEST BINARY FILE
000736  103427                          BCS     30$             ; IF CS, NO ACCESS
000740  110022                          MOVB    R0,(R2)+        ; STORE FUNCTION CODE
000742  116712  000000G                 MOVB    $VALUE,(R2)     ; AND FILE REPEAT COUNT
000746  016746  000000G                 MOV     $SIZE,-(SP)     ; SAVE CURRENT LINE TABLE SIZE
000752                                  CALL    $FILE           ; PROCESS SECONDARY TEMPLATE
000756                                  CALL    $EVEN           ; AUTOMATIC WORD RE-ALIGNMENT
000762  012600                          MOV     (SP)+,R0        ; RESTORE SIZE FROM START OF FILE
000764                                  CALL    $RQ1            ; REQUEST BINARY FILE AGAIN
000770  103412                          BCS     30$             ; IF CS, NO ACCESS
000772  112712  000000G                 MOVB    #.EOF,(R2)      ; STORE END OF FILE FUNCTION CODE
000776  166700  000000G                 SUB     $SIZE,R0        ; GIVING AMOUNT ADDED BY SECONDARY TEMPLATE
001002  000402                          BR      20$
001004  160067  000000G         10$:    SUB     R0,$SIZE        ; ADJUST DATA BASE SIZE
001010  005367  000000G         20$:    DEC     $VALUE          ; BY <FILE COUNT-1> * <FILE SIZE>
001014  001373                          BNE     10$             ;       ...
001016                          30$:    RETURN

                                        .DSABL  LSB
```

```
TMPAC3 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:39  Page 7-1
Symbol table

ASYM    000000R     002 F$$LVL= 000001          L$$DRV= 000000          P$$WRD= 000000          $ERR27  000032R     002
A$$CHK= 000000          F1.STE= ****** GX       L$$P11= 000001          Q$$OPT= 000010          $ERR29  000054R     002
A$$CPS= 000000          GBLK    000006R     002 L$$11R= 000000          REP2A   000210R         $ERR30  000112R     002
A$$PRI= 000000          GETAB   000022R     002 M$$CRB= 000124          R$$DER= 000000          $ERR31  000150R     002
A$$TRP= 000000          GSYM    000012R     002 M$$CRX= 000000          R$$K11= 000001          $ERR32  000176R     002
C$$CKP= 000000          GVAL    000016R     002 M$$FCS= 000000          R$$SND= 000000          $FLAG1= ****** GX
C$$ORE= 000400          G$$TPP= 000000          M$$MGE= 000000          R$$11M= 000000          $FNOUT= ****** GX
C$$RSH= 177564          G$$TSS= 000000          M$$NET= 000000          STBEXT  000304R     002 $GLOBL  000000RG
D$$BUG= 177514          G$$TTK= 000000          M$$OVR= 000000          S$$WRG= 000000          $PRV.N= ****** GX
D$$ISK= 000000          G$$WRD= 000000          N$$ACC= 000001          S$$YSZ= 007600          $REP.C= ****** GX
D$$L11= 000001          I$$RAR= 000000          N$$BUF= 000001          T$$KMG= 000000          $REP.P= ****** GX
D$$YNC= 000000          I$$RDN= 000000          N$$LDV= 000001          T$$MIN= 000000          $SYMBL= ****** GX
D$$YNM= 000000          K$$CNT= 177546          N$$MCP= 000001          V$$CTR= 001000          $TERR = ****** GX
E$$XPR= 000000          K$$CSR= 177546          N$$MLL= 000001          X$$TBT= 000000          $TWARN= ****** GX
FMT2A   000224R     002 K$$LDC= 000000          N$$MOV= 000010          $AZFS = ****** GX       $VALUE= ****** GX
FMT4    000254R     002 K$$TPS= 000074          N$$NCT= 000001          $CAT5 = ****** GX       .PSTCN= ****** GX
FM.2A = 000000          LD$LP = 000000          N$$PEM= 000001          $CLEVL= ****** GX       .PSTPT= ****** GX
FM.4  = 000000          L$$ASG= 000000          P$$P45= 000000          $ELINE= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000270    001   (RW,I,LCL,REL,CON)
DATA    000310    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10034 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:09.31
SY:TMPAC3.V2,[132,134]TMPAC3/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC3
```



```
TMPAC3     CREATED BY  MACRO  ON 29-JUN-85 AT 00:39     PAGE 1
```

```
                                        ;+
                                        ; REP6 - REPLACE PROCESS NAME AND THEN FORM APR NUMBER
                                        ;
                                        ; INPUTS:
                                        ;       R3=FORMAT STRING ADDRESS
                                        ;       R4=ERROR MESSAGE BUFFER ADDRESS
                                        ;       $CAPR=APR NUMBER IN BINARY
                                        ;
                                        ; OUTPUTS:
                                        ;       R0=DESTROYED
                                        ;       R3,R4=UPDATED
                                        ;-
                                        
                                                .IF DF  M$$MGE
                                        
000626                                  REP6:   CALL    $REP.C          ; REPLACE PROCESS NAME
000632  010500                                  MOV     R5,R0           ; COPY ERROR MESSAGE BLOCK ADDRESS
000634  005720                                  TST     (R0)+           ; NOW POINTS AT TEXT STRING
000636  122720  000120                  10$:    CMPB    #'P,(R0)+       ; LOOK FOR "P"
000642  001375                                  BNE     10$             ;       ...
000644  122720  000122                          CMPB    #'R,(R0)+       ; FOLLOWED BY "R"
000650  001372                                  BNE     10$             ;       ...
000652  112710  000060                          MOVB    #'0,(R0)        ; FORM APR NUMBER
000656  156710  000000G                         BISB    $CAPR,(R0)      ;       ...
000662                                          RETURN
                                        
                                                .ENDC
                                        
                                        
000001                                          .END
```

TMPAC5 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:40 Page 8-2
Symbol table

```
A$$CHK= 000000          G$$TSS= 000000          M$$FCS= 000000          Q$$OPT= 000010          $CLDEQ= ****** GX
A$$CPS= 000000          G$$TTK= 000000          M$$MGE= 000000          R$$DER= 000000          $CLIOS= ****** GX
A$$PRI= 000000          G$$WRD= 000000          M$$NET= 000000          R$$K11= 000001          $CLOPE= ****** GX
A$$TRP= 000000          I$$RAR= 000000          M$$OVR= 000000          R$$SND= 000000          $CUR.N= ****** GX
BN.LBL  000162R         I$$RDN= 000000          N$$ACC= 000001          R$$11M= 000000          $EFNLL= ****** GX
BN.OPN  000106R         KISAR0= 172340          N$$BUF= 000001          SWR    = 177570         $ERR69  000000R     002
BUFUMP= 172354          KISAR6= 172354          N$$LDV= 000001          S$$WRG= 000000          $ERR71  000036R     002
CMODE = 140000          K$$CNT= 177546          N$$MCP= 000001          S$$YSZ= 007600          $ERR72  000066R     002
C$$CKP= 000000          K$$CSR= 177546          N$$MLL= 000001          TPS    = 177564         $ERR73  000140R     002
C$$ORE= 000400          K$$LDC= 000000          N$$MOV= 000010          TS$NHD= ****** GX       $ERR74  000202R     002
C$$RSH= 177564          K$$TPS= 000074          N$$NCT= 000001          T$$KMG= 000000          $ERR75  000244R     002
DFAPR = 000006          LD$LP = 000000          N$$PEM= 000001          T$$MIN= 000000          $FILEN= ****** GX
DSKBUF  000000R     003 L$BEXT= ****** GX       PIRQ   = 177772         UBMPR = 170200          $FLAGS= ****** GX
D$$BUG= 177514          L$BFLG= ****** GX       PMODE  = 030000         UISAR0= 177640          $FLAG2= ****** GX
D$$ISK= 000000          L$BLDZ= ****** GX       PR0    = 000000         UISAR1= 177642          $IOSB  = ****** GX
D$$L11= 000001          L$BMXZ= ****** GX       PR1    = 000040         V$$CIR= 001000          $LBN   = ****** GX
D$$YNC= 000000          L$BSA = ****** GX       PR2    = 000100         X$$DBT= 000000          $PRV.N= ****** GX
D$$YNM= 000000          L$$ASG= 000000          PR3    = 000140         $ABFI   000000RG        $REP.C= ****** GX
E$$XPR= 000000          L$$DRV= 000000          PR4    = 000200         $AZFS = ****** GX       $REP.P= ****** GX
FL.CHA= ****** GX       L$$P11= 000001          PR5    = 000240         $BFALL= ****** GX       $RLBL  = ****** GX
FMT6    000310R     002 L$$11R= 000000          PR6    = 000300         $BFBAS= ****** GX       $RQ7   = ****** GX
FM.6   = 000000         MPAR   = 172100         PR7    = 000340         $BFLBN= ****** GX       $TERR  = ****** GX
F$$LVL= 000001          MPCSR  = 177746         PS     = 177776         $BFPTR= ****** GX       $VALUE= ****** GX
F2.FIL= ****** GX       M$$CRB= 000124          P$$P45= 000000          $BFXFR= ****** GX       $VFLV1= ****** GX
G$$TPP= 000000          M$$CRX= 000000          P$$WRD= 000000          $CAPR  = ****** GX       .BASEB= 140000
```

```
. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000346    001   (RW,I,LCL,REL,CON)
DATA    000332    002   (RW,D,LCL,REL,CON)
..BUF   000002    003   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11137 Words  ( 44 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:11.55
SY:TMPAC5./2,[132,134]TMPAC5/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC5
```


TMPAC5 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 1

```
                                                MOV     R1,$FNR50+4     ;
                                                CLR     $LBADR          ; INDICATE NO LIBRARY ADDRESS YET

                                                SWSTK$  60$             ;* ENTER KERNEL MODE
                                                MOV     @NTLPT,R0       ;* GET NETLDR DATA BLOCK BIAS
                                                CALL    $MAPX           ;* MAP TO THE DATA BLOCK

                                                MOV     R2,R0           ;* SAVE VIRTUAL ADDRESS
                                                ADD     #.CXLB1,R2      ;* OFFSET TO LIBRARY DESCRIPTORS
                                                CALL    CHECK           ;* LOOK FOR A MATCH
                                                BCC     50$             ;* IF CC, GOT ONE!
                                                CALL    CHECK           ;* TWO DESCRIPTORS IN DATA BLOCK
                                                BCC     50$             ;*      ...

                                                MOV     .CXLBR(R0),(R1) ;* MAP TO THE FIRST LIBRARY BLOCK
                                                BEQ     50$             ;* IF EQ, THERE IS NONE
                                        40$:    MOV     R0,R2           ;* RESTORE VIRTUAL ADDRESS
                                                CMP     (R2)+,(R2)+     ;* SKIP LINK WORD AND COUNT
                                                CALL    CHECK           ;* LOOK FOR A MATCH
                                                BCC     50$             ;* WITH FIVE DESCRIPTORS IN THIS BLOCK
                                                CALL    CHECK           ;*      ...
                                                BCC     50$             ;*      ...
                                                CALL    CHECK           ;*      ...
                                                BCC     50$             ;*      ...
                                                CALL    CHECK           ;*      ...
                                                BCC     50$             ;*      ...

                                                .IF DF  M$$MGE
                                                MOV     (R0),(R1)       ;* MAP TO THE NEXT LIBRARY BLOCK
                                                BNE     40$             ;* IF NE, THERE IS ANOTHER ONE
                                                .IF DF  R$$11M
                                        50$:    RETURN                  ;* BACK TO USER MODE
                                                .IFF
                                        50$:    CALL    @(SP)+          ;;; RE-ENABLE TASK SWITCHING
                                                .ENDC
                                                .IFF
                                                MOV     (R0),R0         ; GET NEXT LIBRARY BLOCK ADDRESS
                                                BNE     40$             ; IF NE, THERE IS ANOTHER ONE
                                        50$:
                                                .ENDC

000344  032767  000000G 000000G 60$:    BIT     #F2.LBR,$FLAG2  ; WAS LIBRARY FOUND?
000352  001003                          BNE     70$             ; IF NE, YES
000354  062716  000002          65$:    ADD     #2,(SP)         ; REJECT TPARS TRANSITION
000360  000422                          BR      80$
000362                          70$:    CALL    $VFLV1          ; MAKE SURE IT'S THE PRIMARY TEMPLATE
000366                                  CALL    FFN             ; FIX FILE NAME FOR ERROR MESSAGES

000372  012767  140000  000000G        MOV     #.BASEB,$VALUE  ; SET DEFAULT MAPPING
000400  112767  000006  000000G        MOVB    #DFAPR,$CAPR    ; AND DEFAULT APR NUMBER
000406                                  CALL    $RQ7            ; REQUEST BINARY BUFFER ACCESS
000412  010267  000000G                MOV     R2,$BFPTR       ; SAVE BUFFER POINTER
000416  103003                          BCC     80$             ; IF CC, ACCESS GRANTED
000420  042767  000000G 000000G        BIC     #F2.FIL,$FLAG2  ; ELSE CLEAR VALID FILE NAME FLAG
000426                          80$:    RETURN
```

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES	
.PSTCN	= ******	GX	10-199	10-223
.PSTPT	= ******	GX	10-197	
.X2CH1	= ******	GX	15-430	16-472
.X3C01	= ******	GX	15-440	16-482

C 9

```
TMPAC7 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:40  Page 13-1
Symbol table

A$$CHK= 000000          F2.LBR= ****** GX       M$$CRX= 000000          S$$WRG= 000000          $ERRY6  000154R     002
A$$CPS= 000000          G$$TPP= 000000          M$$FCS= 000000          S$$YSZ= 007600          $FILEN= ****** GX
A$$PRI= 000000          G$$TSS= 000000          M$$MGE= 000000          T$$KMG= 000000          $FLAGS= ****** GX
A$$TRP= 000000          G$$TTK= 000000          M$$NET= 000000          T$$MIN= 000000          $FLAG1= ****** GX
BASE    000356R         G$$WRD= 000000          M$$OVR= 000000          USE     000426R         $FLAG2= ****** GX
C$$CKP= 000000          I$$RAR= 000000          NTLPT = ****** GX       V$$CTR= 001000          $LBADR= ****** GX
C$$ORE= 000400          I$$RDN= 000000          N$$ACC= 000001          XFER    000112R         $LBERR  000540RG
C$$RSH= 177564          K$$CNT= 177546          N$$BUF= 000001          X$$DBT= 000000          $LBIAS= ****** GX
D$$BUG= 177514          K$$CSR= 177546          N$$LDV= 000001          $ALB  = ****** GX       $LBXFR= ****** GX
D$$ISK= 000000          K$$LDC= 000000          N$$MCP= 000001          $ALL18= ****** GX       $LIBR   000000RG
D$$L11= 000001          K$$TPS= 000074          N$$MLL= 000001          $ALN18= ****** GX       $MAPX = ****** GX
D$$YNC= 000000          LBLK    000000RG    002 N$$MOV= 000010          $BFALL= ****** GX       $NALL = ****** GX
D$$YNM= 000000          LB.APR= ****** GX       N$$NCT= 000001          $BFBAS= ****** GX       $NLIB = ****** GX
E$$XPR= 000000          LB.USE= ****** GX       N$$PEM= 000001          $BFPTR= ****** GX       $REP.C= ****** GX
FL.CHA= ****** GX       LD$LP = 000000          P$$P45= 000000          $CAPR = ****** GX       $REP.P= ****** GX
FMT2    000234RG    002 LINK    000306R         P$$WRD= 000000          $CLDEQ= ****** GX       $SIZE = ****** GX
FMT6    000253RG    002 L$$ASG= 000000          Q$$OPT= 000010          $DEA18= ****** GX       $TERR = ****** GX
FM.2  = 000000          L$$DRV= 000000          REP6    000574R         $EFNLL= ****** GX       $TWARN= ****** GX
FM.6  = 000000          L$$P11= 000001          R$$DER= 000000          $ELINE= ****** GX       $VALUE= ****** GX
F$$LVL= 000001          L$$11R= 000000          R$$K11= 000001          $ERRY2  000002R     002 .CXLBR= ****** GX
F1.UMR= ****** GX       MAX14   000252R         R$$SND= 000000          $ERRY3  000036R     002 .LIBR = ****** GX
F2.FIL= ****** GX       M$$CRB= 000124          R$$11M= 000000          $ERRY4  000110R     002

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000650    001   (RW,I,LCL,REL,CON)
DATA    000274    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10250 Words  ( 41 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:13.15
SY:TMPAC7.V2,[132,134]TMPAC7/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC7
```

D 9

```
TMPAC7     CREATED BY  MACRO  ON 29-JUN-85 AT 00:41     PAGE 1
```

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  ASR$,EMSG$,NTLER$

                        ;***
                        ; LOCAL DATA
                        ;***

000000                          .PSECT  DATA,D

                        ;
                        ; LIBRARY ADDRESS WITHIN NTPOOL
                        ;
000000                  BFADR:  .BLKW   1

                        ;
                        ; NETLDR DATA BLOCK BIAS
                        ;
000002                  NLBLK:  .BLKW   1

                                .IF DF  R$$MPL

                        LIBTAB: .ASCII  /$XXXVT/                ; LIBRARY VECTOR TABLE START
                        LIBLEN: .ASCII  /$XXXVL/                ; LIBRARY VECTOR TABLE LENGTH

                        STBEXT: .ASCIZ  /STB/                   ; SYMBOL TABLE EXTENSION


                        LIBTB:  .WORD   LIBLN,0                 ; SYMBOL CONTROL BLOCK FOR $XXXVT
                        LIBTB5: .BLKW   2
                        TABVAL: .WORD   0

                        LIBLN:  .WORD   0,0                     ; SYMBOL CONTROL BLOCK FOR $XXXVL
                        LIBLN5: .BLKW   2
                        LENVAL: .WORD   0

                                .ENDC

                        ;
                        ; ERROR MESSAGE (FORMAT STRING ADDRESS DEFINED GLOBALLY IN "TMPAC7")
                        ;
                                .ENABL  LC

000004                          NTLER$  ,Y5,6,$TERR,$CUR.N,$EFNLL,<Read Failure (-***.)>

                                .DSABL  LC

000000                          .PSECT
```

```
                                        .IF DF  M$$MGE

000104  032767  000000G 000000G         BIT     #FL.DDM,$FLAGS          ;IS PROCESS A DDM?
000112  001442                          BEQ     20$                     ;IF EQ, NO
000114  016701  000000G                 MOV     $PDVA,R1                ;GET PDV ADDRESS
000120  032761  000000G 000000G         BIT     #ZF.PSE,Z.FLG(R1)       ;IS THIS A PSEUDO-DEVICE?
000126  001034                          BNE     20$                     ;BR IF YES
000130  005067  000000G                 CLR     $ICBSZ                  ;CLEAR SIZE OF THE ICB ALLOCATION

                                        .IF DF R$$MPL
                                        BIT     #F2.DAS,@FM2TMP         ;IS DATA SPACE ENABLED ?
                                        BNE     13$                     ;IF NE, YES - ALWAYS NEED ICB
                                        .ENDC

                                        .IF DF  R$$11M
000134  020027  120000                  CMP     R0,#120000              ;IS LINE TABLE AT END OF PARTITION?
000140  103427                          BLO     20$                     ; IF LO, NO

                                        .IFF

                                        CMP     R0,#60000               ; IS LINE TABLE AT END OF PARTITION?
                                        BHIS    20$                     ; BR IF NO
                                        .ENDC

                                ;
                                ; ALLOCATE ICB FOR DDM.
                                ;
000142  005001                  13$:    CLR     R1                      ; CLEAR BYTE COUNT NEEDED

                                        .IF DF  R$$11M
000144                                  SWSTK$  18$                     ;* ENTER KERNEL MODE
                                        .ENDC

000150  016700  000000G                 MOV     $NVECT,R0               ;* GET NUMBER OF INT. VETCTORS
000154  066701  000000G         15$:    ADD     $ICBLN,R1               ;* ADD IN SIZE OF EACH INT. VECTOR
000160  005300                          DEC     R0                      ;* DONE?
000162  003374                          BGT     15$                     ;* IF GT, NO
000164  010167  000000G                 MOV     R1,$ICBSZ               ;* STORE SIZE OF REQUIRED ALLOCATION

                                        .IF DF  R$$MPL

                                        ADD     #4,$ICBSZ               ;* DON'T FORGET LINK POINTER AND COUNT FIELD
                                        CALL    $ALICB                  ;* ALLOCATE THE ICB'S
                                                                        ;* NOTE - R0 = ZERO ON MP SYSTEMS

                                        .IFF

                                        .IF DF  R$$11M
000170                                  CALL    @ALOCB                  ;* ALLOCATE FROM RSX POOL.

                                        .IFF
                                        NEG     R1                      ;* DON'T ALLOCATE FROM CEX POOL
                                        CLC                             ;* BE SURE CARRY IS CLEAR
                                        .INH0                           ;* DISABLE TASK SWITCHING
                                        CALL    @ALOCB                  ;* ALLOCATE FROM SCOM
                                        ADC     (SP)                    ;* SAVE CARRY
```

```
                                        .SBTTL  .DEA16 - END OF PROCESS DEALLOCATION
                                ;+
                                ; .DEA16 - END OF PROCESS DE-ALLOCATION ROUTINE
                                ;
                                ; THIS ROUTINE IS CALLED IN PLACE OF "$DEA16", TO ALLOW BOTH NORMAL AND
                                ; PARTITION ALLOCATIONS TO BE RELEASED.
                                ;
                                ; INPUTS:
                                ;       R0 = ALLOCATION ADDRESS
                                ;       R1 = ALLOCATION SIZE
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2 = DESTROYED
                                ;       R3,R4,R5 = SAVED
                                ;-
001372                          .DEA16:
                                        .IF DF  M$$MGE

                                          .IF DF R$$11M

001372  022700  120000                  CMP     #120000,R0      ; IS ALLOCATION FROM PARTITION SPACE ?
001376  101036                          BHI     20$             ; NO .. DE-ALLOCATE NORMALLY

                                          .IFF  ; DF R$$11M

                                        CMP     R0,#10000       ; IS ALLOCATION FROM PARTITION SPACE?
                                        BHIS    20$             ; BR IF NO - DEALLOCATE NORMALLY
                                        SAVRG   <R3,R4,R5>      ; SAVE R3-R5

                                          .IFTF ; DF R$$11M

001400  016704  000000G                MOV     $PDVA,R4        ; GET THE PDV ADDRESS

                                          .IFT  ; DF R$$11M

001404                                  SWSTK$  30$             ;* ENTER KERNEL MODE

                                          .IFTF ; DF R$$11M

001410  010003                          MOV     R0,R3           ;* COPY ALLOCATION ADDRESS
001412  010105                          MOV     R1,R5           ;* AND SIZE

                                          .IFT  ; DF R$$11M

001414  016700  000000G                MOV     $PCB,R0         ;* GET THE PCB ADDRESS
001420  016000  000000G                MOV     P.REL(R0),R0    ;* MAP TO THE PROCESS

                                          .IFF  ; DF R$$11M

                                        MOV     $GCD,R0         ;* GET THE GCD ADDRESS
                                        MOV     G.BA(R0),R0     ;* GET BIAS OF PROCESS

                                          .IFTF ; DF R$$11M

001424                                  CALL    $MAPX           ;*
001430  010300                          MOV     R3,R0           ;* RESTORE ALLOCATION ADDRESS
001432  010501                          MOV     R5,R1           ;* AND SIZE
```

TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASR$	#5-58									
CALL	8-177	8-200	8-237	8-255	8-288	8-301	8-302	9-329	9-331	11-497
	11-512	11-515	11-549	11-581	11-586	11-590	11-597	11-636	11-637	14-721
	14-778	15-820	15-839	15-847	15-879					
CALLR	14-704									
CRASH	9-429	12-661								
EMSG$	#5-58									
EMSG$R	#5-58	8-299	8-300	8-303	9-428	10-473	10-474	10-475	10-476	11-638
NXRDF$	#5-59	5-69								
NTLER$	#5-58	7-103	7-104	7-105	7-106	7-107	7-108	7-109	7-110	7-111
	7-112									
RESRG	#5-59									
RETURN	8-276	8-291	9-334	9-424	10-468	11-559	11-631	12-668	13-694	14-782
	15-858	15-880								
SAVRG	#5-59									
SLTDF$	#5-58	5-61								
SOB	11-596									
SWSTK$	8-237	9-329	11-512	11-586	14-721	15-820				

FILEID**TMPERR

TMPEXP CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL VALUE REFERENCES

```
FL.CHA  = ******  GX    5-76
M$$MGE  = 000000      5-79
R$$11M  = 000000      5-81    5-107
$ADDR   = ******  GX    5-72    5-82    5-108
$FLAGS  = ******  GX    5-76
$FLAG1  = ******  GX    5-74
$NALL   = ******  GX    5-99
$TMEXP    000000 RG   #5-71
$TMEX2  = ******  GX    5-97
$TMEX3  = ******  GX    5-101
$TMEX5  = ******  GX    5-123
$TMEX6  = ******  GX    5-84
$TMEX7  = ******  GX    5-110
```


TMPEXP CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 2

```
                                        ; UNIT NUMBER (BYTE)
                                        ;
000636  005767  000154'         UNB:    TST     FLAGS           ; IS IT EITHER MUX OR MTP EXPANSION?
000642  001403                          BEQ     101$            ; IF EQ, NO
000644  116724  000152'                 MOVB    INDEX,(R4)+     ; STORE CURRENT MUX INDEX
000650                                  RETURN
000652  112724  177777           101$:  MOVB    #-1,(R4)+       ; UNB NOT IN EITHER MXTAB OR MPTAB FILE
000656                                  RETURN

                                        ;
                                        ; UNIT NUMBER (WORD)
                                        ;
000660  005767  000154'         UNW:    TST     FLAGS           ; IS IT EITHER MUX OR MTP EXPANSION?
000664  001403                          BEQ     101$            ; IF EQ, NO
000666  016724  000152'                 MOV     INDEX,(R4)+     ; STORE CURRENT MUX INDEX
000672                                  RETURN
000674  012724  177777           101$:  MOV     #-1,(R4)+       ; UNW NOT IN EITHER MXTAB OR MPTAB FILE
000700                                  RETURN

                                        ;
                                        ; SINGLE WORD
                                        ;
000702  112524                  WORD:   MOVB    (R5)+,(R4)+     ; COPY A WORD
000704  112524                          MOVB    (R5)+,(R4)+
000706                                  RETURN
```

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 1

CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ABORB	000046 R	6-89 #8-195
ADDRW	000072 R	6-90 #8-205
APR	001630 R	6-120 #15-714
BLKB	000122 R	6-91 #8-217
BLKW	000134 R	6-92 #8-226
BYTE	000146 R	6-93 #9-235
CORE	000710 R	6-94 #10-432
CSR	000152 R	6-95 #9-241
CTIM	000164 R	6-96 #9-249
DECPT	= ****** GX	9-300 9-363
DPRB	001656 R	6-121 #15-726
DPRW	001664 R	6-122 #15-732
DVCHA	000172 R	6-97 #9-254
D$LNUM	000014	9-364
D$MLL	000040	9-301
EOF	000342 R	6-98 #9-295
FILE	001200 R	6-99 #13-575
FLAGS	000154 R	#6-151 *7-174 9-259 9-284 9-322 9-340 9-351 9-377 9-390 9-405 9-415 *13-574 *13-590 *13-596 *13-622 *15-674 *15-696 15-714
FL.DDM	= ****** GX	8-195 8-205 9-255 9-280
FL.DLC	= ****** GX	8-195 8-205
FL.KMX	= ****** GX	9-325
FL.MUX	= ****** GX	9-257 9-282 17-799 17-809 17-819 17-829 17-839
INDEX	000152 R	#6-146 *7-173 9-261 9-286 9-324 9-342 9-353 9-381 9-394 9-407 9-417 *13-585 13-586 *13-589 *13-617 13-618 *13-621 *15-689 15-690 *15-695 15-716 17-801 17-811 17-821 17-831 17-841
INT	001122 R	6-100 #12-552
I$$AS	= ******	10-484 10-501 11-542
KSAR5	= ****** GX	12-553 12-556
LB.ADR	= ****** GX	14-652
LFILE	001266 R	6-103 #13-597
LF.ACT	= 100000	#5-64
LF.BRO	= 000400	#5-64
LF.BWT	= 000007	#5-64
LF.ENA	= 002000	#5-64
LF.LPB	= 001000	#5-64
LF.MDC	= 000100	#5-64
LF.MFL	= 004000	#5-64
LF.MTP	= 000020	#5-64
LF.PAC	= 000200	#5-64
LF.RDY	= 040000	#5-64
LF.REA	= 010000	#5-64
LF.SER	= 000040	#5-64
LF.TIM	= 000010	#5-64
LF.UNL	= 020000	#5-64
LF.X2P	= 000000	#5-64
LIBR	001406 R	6-101 #14-628
LINKS	000344 R	6-102 #9-300
LN.CLO	= 000000	#5-64
LN.DUM	= 000005	#5-64
LN.LOA	= 000004	#5-64
LN.LOO	= 000003	#5-64

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 2

Table of contents

5- 55 MACRO DEFINTIONS
6- 68 LOCAL DATA
7- 91 ERROR MESSAGES
8- 112 ACTION ROUTINES

D 2

```
TMPACO - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:38  Page 14-1
Symbol table

A$$CHK= 000000          F1.IN1= ****** GX       M$$FCS= 000000          N$$PEM= 000001          $ERR25  000106R     002
A$$CPS= 000000          F1.IN2= ****** GX       M$$MGE= 000000          P$$P45= 000000          $FLAGS= ****** GX
A$$PRI= 000000          F1.IN3= ****** GX       M$$NET= 000000          P$$WRD= 000000          $FLAG1= ****** GX
A$$TRP= 000000          G$$TPP= 000000          M$$OVR= 000000          Q$$OPT= 000010          $INT    000032RG
C$$CKP= 000000          G$$TSS= 000000          NSITAB  000000R     002 REP2B   000274R     $INTX   000000RG
C$$ORE= 000400          G$$TTK= 000000          NS0    = ****** GX      R$$DER= 000000          $INT1   000054RG
C$$RSH= 177564          G$$WRD= 000000          NS1    = ****** GX      R$$K11= 000001          $INT2   000076RG
D$$BUG= 177514          I$$RAR= 000000          NS2    = ****** GX      R$$SND= 000000          $INT3   000122RG
D$$ISK= 000000          I$$RDN= 000000          NS3    = ****** GX      R$$11M= 000000          $MXVCT= ****** GX
D$$L11= 000001          K$$CNT= 177546          NS4    = ****** GX      S$$WRG= 000000          $NVECT= ****** GX
D$$YNC= 000000          K$$CSR= 177546          NS5    = ****** GX      S$$YSZ= 007600          $REP.C= ****** GX
D$$YNM= 000000          K$$LDC= 000000          NS6    = ****** GX      T$$KMG= 000000          $RQ3  = ****** GX
E$$XPR= 000000          K$$TPS= 000074          NS7    = ****** GX      T$$MIN= 000000          $SIZE = ****** GX
FL.CHA= ****** GX       LD$LP = 000000          N$$ACC= 000001          VFVCT   000146R         $TERR = ****** GX
FMT2    000134RG    002 L$$ASG= 000000          N$$BUF= 000001          V$$CTR= 001000          $VALUE= ****** GX
FMT2B   000153RG    002 L$$DRV= 000000          N$$LDV= 000001          X$$DBT= 000000          $VECT1= ****** GX
FM.2  = 000000          L$$P11= 000001          N$$MCP= 000001          $CBOMG= ****** GX       $VFDDM= ****** GX
FM.2B = 000000          L$$11R= 000000          N$$MLL= 000001          $ELINE= ****** GX       $VFSPC= ****** GX
F$$LVL= 000001          M$$CRB= 000124          N$$MOV= 000010          $ERR23  000020R     002 $$VECT= ****** GX
F1.INT= ****** GX       M$$CRX= 000000          N$$NCT= 000001          $ERR24  000060R     002 .INT  = ****** GX


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000330    001   (RW,I,LCL,REL,CON)
DATA    000204    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10230 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:10.45
SY:TMPACO.V2,[132,134]TMPACO/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPACO
```

E 2

```
TMPACO     CREATED BY  MACRO  ON 29-JUN-85 AT 00:39     PAGE 1
```

```
                                ;+
                                ; $SECSR - ".SECSR" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
001020                          $SECSR::
001020                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
001024                                  CALL    $VFMUX          ; AND THAT IT'S A MUX LINE
001030  032767  000000G 000000G        BIT     #FL.LMC,$FLAGS  ; IS IT A COMIOP DEVICE?
001036  001411                         BEQ     101$            ; IF EQ, NO
001040                                  CALL    $RQ1            ; REQUEST BINARY FILE
001044  103405                         BCS     10$             ; IF CS, NO ACCESS
001046  112712  000000G                MOVB    #.SECSR,(R2)    ; STORE FUNCTION CODE
001052  062767  000002  000000G        ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
001060                          10$:    RETURN
                                ;
                                ; ERROR CONDITION
                                ;
^01062                          101$:   EMSG$R  #5              ; NOT COMIOP DEVICE
```

TMPAC3 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ASYM	000000 R	#5-72 6-139 6-152 7-197
FMT2A	000224 R	5-95 #5-108
FMT4	000254 R	5-96 5-97 5-98 5-99 #5-109
FM.2A =	000000	#5-95
FM.4 =	000000	#5-96 #5-97 #5-98 #5-99
F1.STE =	****** GX	6-136 6-176
GBLK	000006 R	#5-77 6-163
GETAB	000022 R	#5-85 6-177
GSYM	000012 R	#5-78 *6-155 *6-153
GVAL	000016 R	#5-79 6-166
REP2A	000210 R	5-95 #7-192
STBEXT	000304 R	#5-117 6-161
S$$BAS =	******	5-95 5-95 5-96 5-96 5-97 5-97 5-98 5-98 5-99 5-99
$AZFS =	****** GX	6-162
$CAT5 =	****** GX	6-154 6-157
$CLEVL =	****** GX	6-174
$ELINE =	****** GX	5-95
$ERR27	000032 R	5-84 #5-95
$ERR29	000054 R	5-85 #5-96
$ERR30	000112 R	5-86 #5-97
$ERR31	000150 R	5-87 #5-98
$ERR32	000176 R	5-88 #5-99
$FLAG1 =	****** GX	6-136 *6-176
$FNOUT =	****** GX	5-96 5-97 5-98 5-99
$GLOBL	000000 RG	#6-134
$PRV.N =	****** GX	5-96 5-97
$REP.C =	****** GX	7-192
$REP.P =	****** GX	5-98 5-99
$SYMBL =	****** GX	6-164
$TERR =	****** GX	5-96 5-97 5-98 5-99
$TWARN =	****** GX	5-95
$VALUE =	****** GX	*6-135 *6-166
.PSTCN =	****** GX	6-140
.PSTPT =	****** GX	6-138

TMPAC3 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

TMPAC4 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:39 Page 11-1
Symbol table

```
ALLERR  000444R          G$$TTK= 000000          N$$MCP= 000001          S$$YSZ= 007600          $ELINE= ****** GX
A$$CHK= 000000           G$$WRD= 000000          N$$MLL= 000001          TPS   = 177564          $ERR26  000000R     002
A$$CPS= 000000           I$$RAR= 000000          N$$MOV= 000010          T$$KMG= 000000          $ERR76  000044R     002
A$$PRI= 000000           I$$RDN= 000000          N$$NCT= 000001          T$$MIN= 000000          $ERR77  000100R     002
A$$TRP= 000000           KISAR0= 172340          N$$PEM= 000001          UBMPR = 170200          $ERR78  000136R     002
BUFUMP= 172354           KISAR6= 172354          PIRQ  = 177772          UISAR0= 177640          $FLAGS= ****** GX
CMODE = 140000           K$$CNT= 177546          PMODE = 030000          UISAR1= 177642          $FLAG1= ****** GX
C$$CKP= 000000           K$$CSR= 177546          PR0   = 000000          V$$CTR= 001000          $FLAG2= ****** GX
C$$ORE= 000400           K$$LDC= 000000          PR1   = 000040          X$$DBT= 000000          $IOSB = ****** GX
C$$RSH= 177564           K$$TPS= 000074          PR2   = 000100          $ALL16= ****** GX       $LBN  = ****** GX
DFAPR = 000006           LD$LP = 000000          PR3   = 000140          $ALL18= ****** GX       $LLEN = ****** GX
D$$BUG= 177514           L$$ASG= 000000          PR4   = 000200          $ALN18= ****** GX       $NALL = ****** GX
D$$ISK= 000000           L$$DRV= 000000          PR5   = 000240          $BFALL= ****** GX       $NLIB = ****** GX
D$$L11= 000001           L$$P11= 000001          PR6   = 000300          $BFBAS= ****** GX       $READX= ****** GX
D$$YNC= 000000           L$$11R= 000000          PR7   = 000340          $BFLBN= ****** GX       $REP.C= ****** GX
D$$YNM= 000000           MAX14   000574R         PS    = 177776          $BFPTR= ****** GX       $RQ5  = ****** GX
E$$XPR= 000000           MPAR  = 172100          P$$P45= 000000          $BFXFR= ****** GX       $RQ7  = ****** GX
FL.CHA= ****** GX        MPCSR = 177746          P$$WRD= 000000          $BINF   000000RG        $SCOM   000452RG
FMT2    000210R     002  M$$CRB= 000124          Q$$OPT= 000010          $CAPR = ****** GX       $SIZE = ****** GX
FMT6    000235R     002  M$$CRX= 000000          REP6    000626R         $CLDEQ= ****** GX       $TERR = ****** GX
FM.2  = 000000           M$$FCS= 000000          R$$DER= 000000          $CLIOS= ****** GX       $TWARN= ****** GX
FM.6  = 000000           M$$MGE= 000000          R$$K11= 000001          $CORE   000306RG        $VALUE= ****** GX
F$$LVL= 000001           M$$NET= 000000          R$$SND= 000000          $CORE1  000236RG        $VFLV1= ****** GX
F1.UMR= ****** GX        M$$OVR= 000000          R$$11M= 000000          $CUR.N= ****** GX       .BASEB= 140000
F2.FIL= ****** GX        N$$ACC= 000001          SWR   = 177570          $CVAL = ****** GX       .CORE = ****** GX
G$$TPP= 000000           N$$BUF= 000001          S$$WRG= 000000          $EFNLL= ****** GX       .SCOM = ****** GX
G$$TSS= 000000           N$$IDV= 000001

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000664    001   (RW,I,LCL,REL,CON)
DATA    000256    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11161 Words  ( 44 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:13.13
SY:TMPAC4.V2,[132,134]TMPAC4/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC4
```


TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

TMPAC5 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL VALUE REFERENCES

```
BN.LBL    000162 R    6-153   #8-201
BN.OPN    000106 R    6-151   #7-169
BUFUMP  = 172354      #5-64
CMODE   = 140000      #5-64
DFAPR   = 000006      #5-104   6-135
DSKBUF    000000 R    #5-112   8-201
FL.CHA  = ******  GX  6-149
.MT6      000310 R    5-77     5-78     5-79     5-80     5-82     5-86     #5-95
FM.0    = 000000      #5-77    #5-78    #5-79    #5-80    #5-82    #5-86
F2.FIL  = ******  CX  6-130    6-156
I$$AS   = ******      5-64     8-231
KISAR0  = 172340      #5-64
KISAR6  = 172354      #5-64
L$BEXT  = ******  GX  8-228
L$BFLG  = ******  GX  8-209
L$BLDZ  = ******  GX  8-216
L$BMXZ  = ******  GX  8-217
L$BSA   = ******  GX  8-208
MPAR    = 172100      #5-64
MPCSR   = 177746      #5-64
M$$MGE  = 000000      5-64     5-103    6-133    6-140    8-234
PIRQ    = 177772      #5-64
PMODE   = 030000      #5-64
PR0     = 000000      #5-64
PR1     = 000040      #5-64
PR2     = 000100      #5-64
PR3     = 000140      #5-64
PR4     = 000200      #5-64
PR5     = 000240      #5-64
PR6     = 000300      #5-64
PR7     = 000340      #5-64
PS      = 177776      #5-64
R$$EIS  = ******      8-231
R$$MPL  = ******      5-64     5-64     5-64
R$$11D  = ******      5-64     8-231
R$$11M  = 000000      5-81     8-207    8-220
SWR     = 177570      #5-64
S$$BAS  = ******      5-77     5-78     5-78     5-79     5-79     5-80     5-80     5-82
                      5-82     5-86     5-86
TPS     = 177564      #5-64
TS$NHD  = ******  GX  8-209
UBMPR   = 170200      #5-64
UISAR0  = 177640      #5-64
UISAR1  = 177642      #5-64
$ABFI     000000 RG   #6-129
$AZFS   = ******  GX  7-172
$BFALL  = ******  GX  *6-148   *8-238
$BFBAS  = ******  GX  *8-208
$BFLBN  = ******  GX  *8-204   *8-205
$BFPTR  = ******  GX  *6-147
$BFXFR  = ******  GX  *8-232
$CAPR   = ******  GX  *6-135
```


TMPAC5 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 2

```
                                ;
                                ; FIX FILE NAME FOR .BIN/.LIBR ERROR MESSAGES
                                ;
000430  005000                FFN:    CLR     R0              ; RESET FILE NAME
000432  012701  0000006               MOV     #$FILEN,R1      ; FOR .BIN/.LIBR ERROR MESSAGES
000436  005002                        CLR     R2              ;       ...
000440                                CALLR   $AZFS           ;       ...
```

TMPAC6 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 4

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	9-163	9-171	10-228	10-230	10-232	10-236	10-238	10-242	10-244	10-251
	10-253	10-255	10-257	10-259	10-280	10-281	10-285	12-355	12-356	13-383
	13-384	13-391	15-437	15-445	15-446	15-449	16-479	16-487	16-488	16-491
	17-511	18-529								
CALLR	10-297									
EMSG$	#5-58	9-163	13-391	15-445	16-487	17-511	18-529			
NHWDF$	#5-58	5-62								
NTLER$	#5-58	8-117	8-118	8-120	8-121	8-122				
RESRG	#5-58									
RETURN	9-172	10-266	10-289	11-342	12-360	13-387	13-393	14-410	15-452	16-494
	17-512	18-530								
SAVRG	#5-58									
SOB	10-216	10-226								
SWSTK$	10-236	13-383								

FILEID**TMPAC7

TMPAC7 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES						
BASE	000356 R	6-137	#10-266					
FL.CHA	= ****** GX	6-120	11-314					
FMT2	000234 RG	5-86	5-87	#5-96				
FMT6	000253 RG	5-78	5-83	#5-97				
FM.2	= 000000	#5-86	#5-87					
FM.6	= 000000	#5-78	#5-83					
F1.UMR	= ****** GX	7-176						
F2.FIL	= ****** GX	6-118						
F2.LBR	= ****** GX	6-123	12-349					
LBLK	000000 RG	#5-70	*6-122	6-131	9-243	9-249	12-352	
LB.APR	= ****** GX	10-274						
LB.USE	= ****** GX	*11-322						
LINK	000306 R	6-133	#9-235					
MAX14	000252 R	6-140	7-171	#8-213				
M$$MGE	= 000000	5-77	6-136	7-157	7-175	7-191	10-264	11-302	11-310	12-355
		13-377						
NTLPT	= ****** GX	9-237						
REP6	000574 R	5-78	#13-379					
R$$11M	= 000000	5-82	7-158	7-191	10-268	10-276	11-317	11-323	
S$$BAS	= ******	5-78	5-78	5-83	5-83	5-86	5-86	5-87	5-87
USE	000426 R	6-142	#11-299					
XFER	000112 R	6-129	#7-156					
$ALB	= ****** GX	6-127						
$ALL18	= ****** GX	7-182						
$ALN18	= ****** GX	7-178						
$BFALL	= ****** GX	7-173	7-197					
$BFBAS	= ****** GX	7-159	7-161	*10-274	10-282			
$BFPTR	= ****** GX	11-299						
$CAPR	= ****** GX	10-282	13-393					
$CLDEQ	= ****** GX	12-351						
$DEA18	= ****** GX	7-198	12-361					
$EFNLL	= ****** GX	5-78	5-83					
$ELINE	= ****** GX	5-86	5-87					
$ERRY2	000002 R	#5-78	7-170	10-284				
$ERRY3	000036 R	#5-83	7-193					
$ERRY4	000110 R	#5-86	7-196					
$ERRY6	000154 R	#5-87	8-224					
$FILEN	= ****** GX	13-382						
$FLAGS	= ****** GX	6-120	11-314					
$FLAG1	= ****** GX	7-176						
$FLAG2	= ****** GX	6-118	6-123	12-349				
$LBADR	= ****** GX	6-125	10-273	11-307	11-308	11-321		
$LBERR	000540 RG	7-192	7-195	7-199	8-223	#12-348		
$LBIAS	= ****** GX	10-266	11-303	11-304	11-316			
$LBXFR	= ****** GX	7-184						
$LIBR	000000 RG	#6-117						
$MAPX	= ****** GX	9-239	10-272	11-320				
$NALL	= ****** GX	8-213						
$NLIB	= ****** GX	8-214	*11-338					
$REP.C	= ****** GX	5-86	5-87					
$REP.P	= ****** GX	5-83						
$SIZE	= ****** GX	*11-313						

TMPAC7 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

```
                                        ;+
                                        ; $LBXFR - TRANSFER LIBRARY FILE INTO CORE
                                        ;
                                        ; INPUTS:
                                        ;       R0    = LIBRARY BIAS
                                        ;       $BFLBN=LOGICAL BLOCK NUMBER
                                        ;       $BFXFR=TRANSFER SIZE IN BYTES
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT=SUCCESS/FAILURE
                                        ;-
000000                                  $LBXFR::
000000  010067  000000'                         MOV     R0,BFADR        ; SAVE LIBRARY ADDRESS
000004  016767  000000G 000000G                 MOV     $BFLBN,$LBN     ; COPY LOGICAL BLOCK NUMBER
000012  016767  000002G 000002G                 MOV     $BFLBN+2,$LBN+2 ;   ...
000020  016767  000000G 000000G                 MOV     $BFXFR,$LLEN    ; AND TRANSFER SIZE IN BYTES
000026                                          CALL    $READX          ; READ INTO EXTENDED MEMORY
000032  103446                                  BCS     101$            ; IF CS, READ FAILURE
000034                                          CALL    $CLDEQ          ; CLOSE LIBRARY FILE

                                                .IF DF  R$$MPL

                                                MOVB    $LIBNM,LIBTAB+1 ; COPY FIRST THREE CHARACTERS OF LIBRARY NAME
                                                MOVB    $LIBNM,LIBLEN+1 ; ...
                                                MOVB    $LIBNM+1,LIBTAB+2 ; ...
                                                MOVB    $LIBNM+1,LIBLEN+2 ; ...
                                                MOVB    $LIBNM+2,LIBTAB+3 ; ...
                                                MOVB    $LIBNM+2,LIBLEN+3 ; ...

                                                MOV     #LIBTAB,R0      ; SET UP SYMBOL NAME
                                                CALL    $CAT5           ; CONVERT TO RAD50
                                                MOV     R1,LIBTB5       ; STORE VALUE
                                                CALL    $CAT5           ; CONVERT TO RAD50
                                                MOV     R1,LIBTB5+2     ; STORE RAD50 VALUE OF LAST 3 CHARACTERS
                                                MOV     #LIBLEN,R0      ; SET UP SYMBOL NAME
                                                CALL    $CAT5           ; CONVERT TO RAD50
                                                MOV     R1,LIBLN5       ; STORE RAD50 VALUE
                                                CALL    $CAT5           ; CONVERT TO RAD50
                                                MOV     R1,LIBLN5+2     ; STORE RAD50 VALUE OF LAST 3 CHARACTRES

                                                MOV     #$CXUIC,R0      ; GET DEFAULT UIC
                                                MOV     #$LIBNM,R1      ; GET LIBRARY FILENAME
                                                MOV     #STBEXT,R2              ; GET FILE EXTENSION
                                                CALL    $AZFS           ; SET UP FILE SPEC

                                                MOV     #LIBTB,R0       ; GET START OF SYMBOL BLOCKS
                                                CALL    $SYMBL          ; LOOKUP SYMBOLS
                                                BCS     101$

                                                MOV     LENVAL,$LBVL    ; COPY $XXXVL VALUE
                                                BEQ     5$              ; IF NOT VECTORED - BRANCH
                                                MOV     TABVAL,$LBVT    ; COPY $XXXVT VALUE
                                                BIC     #160000,$LBVT   ; CLEAR APR BITS
                                                BIS     #140000,$LBVT   ; MAP THRU APR6
                                                MOV     BFADR,$LBBLK    ; COPY LIBRARY BIAS
                                                CALL    $VCTLB          ; SET UP LIBRARY VECTOR TABLE
```

```
                                        .ENBO                           ;* RE-ENABLE TASK SWITCHING
                                        .ENDC
                                                .ENDC
                                        .IF DF  R$$11M
000174  103005                         BCC     16$                     ;* IF CC, GOT IT
000176  017701  000000G                 MOV     @HEADR,R1               ;* GET USER MODE SP
000202  011101                         MOV     (R1),R1                 ;* ...
000204  005261  000006                  INC     6(R1)                   ;* SET USER MODE C-BIT
000210  010067  000000G         16$:    MOV     R0,$ICBAD               ;* STORE ADDRESS OF ALLOCATION
000214                                  RETURN                          ;* RETURN TO USER MODE
                                        .IFTF
000216  103420                 18$:    BCS     121$                    ; IF CS, ALLOCATION FAILURE
                                        .IFF
                                        MOV     R0,$ICBAD               ; STORE ADDRESS OF ALLOCATION
                                        .ENDC
                                        .ENDC
000220                          20$:    CALL    UBA                     ;CALCULATE UNLOAD BLOCK ADDRESS
000224                          30$:
000224                          RTSPC:  RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000226  012767  177777  000000G 101$:   MOV     #-1,$ADDR               ;SET ADDRESS FOR EMPTY LLC DATA BASE
000234  032767  000000G 000000G         BIT     #FL.LLC,$FLAGS          ;IS THIS AN LLC?
000242  001370                         BNE     30$                     ;IF NE, YES
000244                                  EMSG$R  33                      ;LINE TABLE EMPTY
000252                          111$:   EMSG$R  34                      ;LINE TABLE ALLOCATION FAILURE
000260                          121$:   CALL    XICB                    ;DEALLOCATE ANY ALLOCATED ICB'S
000264                                  CALL    XLT                     ;RELEASE THE LINE TABLE ALLOCATION
000270                                  EMSG$R  9B                      ;ICB ALLOCATION FAILURE
                                        .DSABL  LSB
```

```
    842 001434  012703  177776G                 MOV     #$PRAVL-2,R3    ;* POINT AT COMMON LISTHEAD
    843 001440  062704  000000G                 ADD     #Z.AVL,R4       ;* POINT AT PROCESS ALLOCATION WORD
    844 001444  011463  000002                  MOV     (R4),2(R3)      ;* SETUP PARTITION FREE SPACE POINTER
    845 001450  042700  160000                  BIC     #160000,R0      ;* CLEAR GIVEN APR
    846 001454  050200                          BIS     R2,R0           ;* USE APR FROM MAPX
    847 001456                                  CALL    $DEAC2          ;* DE-ALLOCATE BACK TO PARTITION SPACE
    848 001462  016714  000000G                 MOV     $PRAVL,(R4)     ;* RESTORE PCB ALLOCATION WORD
    849 001466  042714  160000                  BIC     #160000,(R4)    ;*      ...
    850
    851                                           .IFF  ; DF R$$11M
    852
    853                                         CALL    @(SP)+          ;* CALL CO-ROUTINE TO RESTORE MAPPING
    854                                         RESRG   <R5,R4,R3>      ;* RESTORE REGISTERS
    855
    856                                           .ENDC ; DF R$$11M
    857
    858 001472                                  RETURN                  ;* BACK TO USER MODE
    859                                         .IFF
    860                                         CMP     @EXSIZ,R0       ; IS ALLOCATION FROM RSX POOL ?
    861                                         BHI     20$             ; YES .. DE-ALLOCATE NORMALLY
    862                                         JSR     R5,$SAVRG       ; SAVE R3,R4,R5
    863                                         MOV     @NTLPT,R2       ; GET NTL DATA BLOCK ADDRESS
    864                                         MOV     .CXPCB(R2),R2   ; COMM EXEC PCB ADDRESS
    865                                         MOV     P.REL(R2),R3    ; AND COMM EXEC STARTING ADDRESS
    866                                         CMP     R0,R3           ; IS ALLOCATION FROM CEX POOL ?
    867                                         BLO     10$             ; NO .. IT'S FROM PARTITION SPACE
    868                                         ADD     P.SIZE(R2),R3   ; GET COMM EXEC ENDING ADDRESS
    869                                         CMP     R0,R3           ; IS ALLOCATION FROM CEX POOL ?
    870                                         BLO     20$             ; YES .. DE-ALLOCATE NORMALLY
    871                                 10$:    MOV     $PDVA,R5        ; GET PDV ADDRESS
    872                                         MOV     #$PRAVL-2,R3    ; POINT AT COMMON LISTHEAD
    873                                         ADD     #Z.AVL,R5       ; POINT AT PROCESS ALLOCATION WORD
    874                                         MOV     (R5),2(R3)      ; SETUP PARTITION FREE SPACE POINTER
    875                                         CALL    @DEAC1          ; DE-ALLOCATE BACK TO PARTITION SPACE
    876                                         MOV     $PRAVL,(R5)     ; RESTORE PCB ALLOCATION WORD
    877                                         BR      30$
    878                                         .ENDC
    879 001474                          20$:    CALL    $DEA16          ; DE-ALLOCATE NORMALLY
    880 001500                          30$:    RETURN
```

```
**FILE**ID**TMPERR
```

```
TTTTTTTTTT   MM      MM   PPPPPPPP     EEEEEEEEEE   RRRRRRRR     RRRRRRRR
TTTTTTTTTT   MM      MM   PPPPPPPP     EEEEEEEEEE   RRRRRRRR     RRRRRRRR
    TT       MMMM  MMMM   PP      PP   EE           RR      RR   RR      RR
    TT       MMMM  MMMM   PP      PP   EE           RR      RR   RR      RR
    TT       MM  MM  MM   PP      PP   EE           RR      RR   RR      RR
    TT       MM  MM  MM   PP      PP   EE           RR      RR   RR      RR
    TT       MM      MM   PPPPPPPP     EEEEEEEE     RRRRRRRR     RRRRRRRR
    TT       MM      MM   PPPPPPPP     EEEEEEEE     RRRRRRRR     RRRRRRRR
    TT       MM      MM   PP           EE           RR  RR       RR  RR
    TT       MM      MM   PP           EE           RR  RR       RR  RR
    TT       MM      MM   PP           EE           RR    RR     RR    RR       ....
    TT       MM      MM   PP           EE           RR    RR     RR    RR       ....
    TT       MM      MM   PP           EEEEEEEEEE   RR      RR   RR      RR     ....
    TT       MM      MM   PP           EEEEEEEEEE   RR      RR   RR      RR     ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

TMPEXP CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE ?

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME REFERENCES

CALL 5-84 5-97 5-101 5-110 5-123
RETURN 5-124

FILEID**TMPEX2

```
                                ;
                                ; POINTER TO ALLOCATED CORE
                                ;
000710  112524                  CORE:   MOVB    (R5)+,(R4)+     ; COPY FIRST WORD OF ADDRESS
000712  112524                          MOVB    (R5)+,(R4)+     ;       ...
                                
                                        .IF DF  M$$MGE
                                
000714  112524                          MOVB    (R5)+,(R4)+     ; COPY SECOND WORD OF ADDRESS
000716  112524                          MOVB    (R5)+,(R4)+     ;       ...
                                
                                        .IFTF
                                
000720  112546                          MOVB    (R5)+,-(SP)     ; ASSEMBLE COUNT
000722  112566  000001                  MOVB    (R5)+,1(SP)     ;       ...
000726  012603                          MOV     (SP)+,R3        ;       ...
000730  001436                          BEQ     10$             ; IF ZERO, NO UNLOAD DATA TO STORE
                                
                                        .IFT
                                
                                                .IF DF  R$$11M
                                
000732                                  SWSTK$  10$             ;* ENTER KERNEL MODE
                                
                                                .ENDC
                                
000736  016700  000000G                 MOV     $UBIAS,R0       ;* GET UNLOAD BLOCK BIAS
000742                                  CALL    $MAPX           ;* MAP TO UNLOAD BLOCK
                                
                                        .IFTF
                                
000746  016702  000000G                 MOV     $UDA,R2         ;* GET UNLOAD DATA ADDRESS
                                
                                        .IFT
                                
000752  016422  177774                  MOV     -4(R4),(R2)+    ;* STORE ALLOCATION BIAS
                                
                                        .IFF
                                
                                        MOV     -2(R4),(R2)+    ; STORE ALLOCATION ADDRESS
                                
                                        .IFTF
                                
000756  006303                          ASL     R3              ;* SHIFT COUNT LEFT 1 PLACE
000760  005203                          INC     R3              ;* MAKE IT ODD
000762  010322                          MOV     R3,(R2)+        ;* STORE COUNT
000764  010267  000000G                 MOV     R2,$UDA         ;* SET NEW UNLOAD DATA ADDRESS
                                
                                        .IFT
                                
000770  105765  177771                  TSTB    -7(R5)          ;* IS THIS .CORE OR .BIN?
000774  100014                          BPL     10$             ;* IF PL, .BIN
000776  016411  177774                  MOV     -4(R4),(R1)     ;* MAP TO THE .CORE ALLOCATION
001002  042702  017777                  BIC     #17777,R2       ;* GIVING VIRTUAL ADDRESS
001006  005303                          DEC     R3              ;* CONVERT TO A WORD COUNT
001010                                  ASL$    4,R3            ;*       ...
```

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
LN.OAU	= 000003		#5-64			
LN.OFF	= 000001		#5-64			
LN.ON	= 000000		#5-64			
LN.OOP	= 000004		#5-64			
LN.OPE	= 000001		#5-64			
LN.REF	= 000002		#5-64			
LN.SER	= 000002		#5-64			
LN.STA	= 000017		#5-64			
LN.SUB	= 000360		#5-64			
LN.TRI	= 000006		#5-64			
LSTHD	000360	R	6-104	#9-308		
LTAB	000370	R	6-105	#9-316		
L.COST	000015		#5-64			
L.CTL	000012		#5-64			
L.CVA	177776		#5-64	*9-267		
L.DDM	000002		#5-64			
L.DDS	000004		#5-64			
L.DLC	000003		#5-64			
L.DLM	000006		#5-64			
L.DLS	000010		#5-64			
L.FLG	000000		#5-64			
L.KRBA	000016		#5-64			
L.LEN	= 000022		#5-64			
L.MPF	000022		#5-64			
L.NMST	000020		#5-64			
L.NSTA	000014		#5-64			
L.OWNR	000021		#5-64			
L.UNT	000013		#5-64			
MPTAB	001260	R	6-106	#13-596		
MXPTB	001500	R	6-124	#15-674		
MXTAB	001172	R	6-107	#13-574		
M$$MGE	= 000000		10-435	11-524	14-628	
NOD	000520	R	6-108	#9-363		
PECHA	000274	R	6-125	#9-279		
PRI	001160	R	6-109	#12-567		
PRIOV	000532	R	6-110	#9-371		
PR7	= ******	GX	12-567			
R$$EIS	= ******		10-484			
R$$11D	= ******		10-484	10-501	11-542	
R$$11M	= 000000		10-449	11-526	14-645	
SAVALC	001066	R	#11-524	15-698		
SCOM	001030	R	6-111	#11-511		
SECSR	000376	R	6-112	#9-322		
SF.ACT	= 000200		#5-64			
SF.ENA	= 000100		#5-64			
SF.LPB	= 000004		#5-64			
SF.MFL	= 000040		#5-64			
SF.PAC	= 000020		#5-64			
SF.REA	= 000010		#5-64			
SF.SER	= 000001		#5-64			
SF.SVC	= 000002		#5-64			
SF.UNL	= 000040		#5-64			

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 3

```
                        .TITLE  TMPACO - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 7
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-APR-81
;       DECNET-11M/S V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
;
;
```

TMPACO CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES					
FL.CHA	= ****** GX	13-221					
FMT2	000134 RG	7-100	#7-104				
FMT2B	000153 RG	7-101	7-102	#7-105			
FM.2	= 000000	#7-100					
FM.2B	= 000000	#7-101	#7-102				
F1.INT	= ****** GX	9-144	10-162	11-179	12-196		
F1.IN1	= ****** GX	9-145	10-161				
F1.IN2	= ****** GX	9-145	11-178				
F1.IN3	= ****** GX	9-145	12-195				
I$$AS	= ******	13-226					
NSITAB	000000 R	#6-75	13-242				
NS0	= ****** GX	6-75					
NS1	= ****** GX	6-76					
NS2	= ****** GX	6-77					
NS3	= ****** GX	6-78					
NS4	= ****** GX	6-79					
NS5	= ****** GX	6-80					
NS6	= ****** GX	6-81					
NS7	= ****** GX	6-82					
REP2B	000274 R	7-101	7-102	#14-277			
R$$EIS	= ******	13-226					
R$$MPL	= ******	5-62	6-85	13-229	13-247		
R$$11D	= ******	13-226					
R$$11M	= 000000	6-74					
S$$BAS	= ******	7-100	7-100	7-101	7-101	7-102	7-102
VFVCT	000146 R	9-147	10-164	11-181	12-198	#13-212	
$CBOMG	= ****** GX	14-285					
$ELINE	= ****** GX	7-100	7-101	7-102			
$ERR23	000020 R	#7-100	13-261				
$ERR24	000060 R	#7-101	13-262				
$ERR25	000106 R	#7-102	13-263				
$FLAGS	= ****** GX	13-221					
$FLAG1	= ****** GX	13-214					
$INT	000032 RG	#9-142					
$INTX	000000 RG	#8-122					
$INT1	000054 RG	#10-159					
$INT2	000076 RG	#11-176					
$INT3	000122 RG	#12-193					
$MXVCT	= ****** GX	13-219					
$NVECT	= ****** GX	*13-255					
$REP.C	= ****** GX	7-100	14-277				
$RQ3	= ****** GX	8-123					
$SIZE	= ****** GX	*8-128	13-216				
$TERR	= ****** GX	7-100	7-101	7-102			
$VALUE	= ****** GX	8-126	8-127				
$VECT1	= ****** GX	*13-216					
$VFDDM	= ****** GX	9-143	10-160	11-177	12-194		
$VFSPC	= ****** GX	13-212					
$$VECT	= ****** GX	13-218					
.INT	= ****** GX	8-125					

TMPACO CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

```
                                ;+
                                ; $VFMUX - VERIFY MULTIPLEXED LINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ERROR UNLESS MULTIPLEXED LINE
                                ;-
001070                          $VFMUX::
001070  032767  000000G 000000G         BIT     #FL.MUX,$FLAGS  ; MUX LINE?
001076  001003                          BNE     10$             ; IF NE, YES
001100                                  EMSG$   81              ; NOT MULTIPLEXED LINE
001106                          10$:    RETURN
                                
000001                                  .END
```


Symbol table

TMPAC3 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES					
CALL	6-154	6-157	6-162	6-164	7-192	
CALLR	6-178					
NTLER$	#5-59	5-95	5-96	5-97	5-98	5-99
RETURN	6-169	7-205				
SOB	6-146	6-151	7-202			

FILEID**TMPAC4

TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
ALLERR	000444 R	8-264 #8-295 9-324
BUFUMP	= 172354	#5-64
CMODE	= 140000	#5-64
DFAPR	= 000006	#5-76 7-217
FL.CHA	= ****** GX	6-127 8-253 9-313
FMT2	000216 R	5-84 5-91 #5-100
FMT6	000235 R	5-86 5-90 #5-101
FM.2	= 000000	#5-84 #5-91
FM.6	= 000000	#5-86 #5-90
F1.UMR	= ****** GX	6-147 8-262
F2.FIL	= ****** GX	6-124
I$$AS	= ******	5-64 7-214
KISAR0	= 172340	#5-64
KISAR6	= 172354	#5-64
MAX14	000574 R	6-142 8-258 9-318 #10-346
MPAR	= 172100	#5-64
MPCSR	= 177746	#5-64
M$$MGE	= 000000	5-64 5-75 5-85 6-130 6-146 6-162 6-170 7-207 8-245 8-261 8-276 8-284 11-372
PIRQ	= 177772	#5-64
PMODE	= 030000	#5-64
PR0	= 000000	#5-64
PR1	= 000040	#5-64
PR2	= 000100	#5-64
PR3	= 000140	#5-64
PR4	= 000200	#5-64
PR5	= 000240	#5-64
PR6	= 000300	#5-64
PR7	= 000340	#5-64
PS	= 177776	#5-64
REP6	000626 R	5-86 #11-374
R$$EIS	= ******	7-214
R$$MPL	= ******	5-64 5-64 5-64
R$$11D	= ******	5-64 7-214
R$$11M	= 000000	6-131
SWR	= 177570	#5-64
S$$BAS	= ******	5-84 5-84 5-86 5-86 5-90 5-90 5-91 5-91
TPS	= 177564	#5-64
UBMPR	= 170200	#5-64
UISAR0	= 177640	#5-64
UISAR1	= 177642	#5-64
$ALL16	= ****** GX	9-323
$ALL18	= ****** GX	6-153 8-268
$ALN18	= ****** GX	6-149 8-264
$BFALL	= ****** GX	6-144 6-167 6-168
$BFBAS	= ****** GX	6-132
$BFLBN	= ****** GX	6-178 6-179
$BFPTR	= ****** GX	6-156
$BFXFR	= ****** GX	6-180
$BINF	000000 RG	#6-123
$CAPR	= ****** GX	*7-217 11-382
$CLDEQ	= ****** GX	6-185 6-191

TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

TMPAC5 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES					
$CLDEQ	= ******	GX	6-155	7-184				
$CLOS	= ******	GX	7-174	8-244				
$CLOPE	= ******	GX	7-173					
$CUR.N	= ******	GX	5-79					
$EFNLL	= ******	GX	5-77	5-78	5-79	5-80	5-82	5-86
$ERR69	000000	R	#5-77	7-183				
$ERR71	000036	R	#5-78	7-185				
$ERR72	000066	R	#5-79	8-245				
$ERR73	000140	R	#5-80	8-246				
$ERR74	000202	R	#5-82	8-247				
$ERR75	000244	R	#5-86	8-248				
$FILEN	= ******	GX	7-170					
$FLAGS	= ******	GX	6-149					
$FLAG2	= ******	GX	6-130	*6-156				
$IOSB	= ******	GX	8-244					
$LBN	= ******	GX	8-204	8-205				
$PRV.N	= ******	GX	5-77					
$REP.C	= ******	GX	5-80	5-82	5-86			
$REP.P	= ******	GX	5-78					
$RLBL	= ******	GX	8-202					
$RQ7	= ******	GX	6-141					
$TERR	= ******	GX	5-77	5-78	5-79	5-80	5-82	5-86
$VALUE	= ******	GX	*6-134					
$VFLVT	= ******	GX	6-138					
.BASEB	= 140000		#5-64	5-104	6-134			

TMPAC5 CREATED BY MACRO ON 29-JUN-85 AT 00:4[illegible] PAGE 3

```
                                        .SBTTL  CHECK - CHECK LIBRARY DESCRIPTOR FOR MATCH
                                ;+
                                ; CHECK - CHECK THIS LIBRARY DESCRIPTOR FOR A MATCH
                                ;
                                ; INPUTS:
                                ;       (R1)=LIBRARY BLOCK BIAS (MAPPED ONLY)
                                ;       R2=LIBRARY BLOCK VIRTUAL ADDRESS
                                ;       $FNR50=RAD50 FILE (LIBRARY) NAME TO BE MATCHED
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R2 INCREMENTED BY 12. (SIZE OF LIBRARY DESCRIPTOR)
                                ;-
000444                          CHECK:  .IF DF  M$$MGE

000444  011167  000000G                MOV     (R1),$LBIAS     ;* SAVE BIAS OF THIS BLOCK

                                        .ENDC

000450  005712                         TST     (R2)            ;* IS THIS AN EMPTY SLOT?
000452  001436                         BEQ     10$             ;* IF EQ, YES - SAVE ITS ADDRESS
000454  021267  000000G                CMP     (R2),$FNR50     ;* COMPARE THREE WORDS
000460  001035                         BNE     20$             ;*      ...
000462  026267  000002  000002G        CMP     2(R2),$FNR50+2  ;*      ...
000470  001031                         BNE     20$             ;*      ...
000472  026267  000004  000004G        CMP     4(R2),$FNR50+4  ;*      ...
000500  001025                         BNE     20$             ;*      ...

000502  032767  000000G 000042'        BIT     #FL.CHA,.FLAGS  ;* MUX CHARACTERISTICS UPDATE ONLY ?
000510  001014                         BNE     8$              ;* YES ..
000512  032767  000000G 000000G        BIT     #F1.UMR,$FLAG1  ;* MUST FILE BE IN UMR SPACE?
000520  001404                         BEQ     5$              ;* BR IF NO, DOESN'T MATTER
000522  026277  000000G 000000G        CMP     LB.ADR(R2),@QSTRT ;* IS THIS COPY IN UMR SPACE
000530  103411                         BLO     20$             ;* BR IF NO, CAN'T USE IT
000532  126227  000000G 000377  5$:     CMPB    LB.USE(R2),#377 ;* IS USE COUNT AT MAX VALUE?
000540  001405                         BEQ     20$             ;* IF EQ, YES - CAN'T USE THIS ONE

000542  052767  000000G 000000G 8$:     BIS     #F2.LBR,$FLAG2  ;* WE FOUND IT!
000550  010267  000000G         10$:    MOV     R2,$LBADR       ;* SAVE ADDRESS OF THIS DESCRIPTOR
000554  062702  000014          20$:    ADD     #12.,R2         ;* ADVANCE R2 TO NEXT DESCRIPTOR (CLEARS C-BIT)
000560  032767  000000G 000000G        BIT     #F2.LBR,$FLAG2  ;* DID WE FIND A MATCH?
000566  001001                         BNE     30$             ;* IF NE, YES
000570  000261                         SEC                     ;* INDICATE FAILURE
000572                          30$:    RETURN                  ;*
```

```
**FILE**ID**TMPAC7
```

```
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC   77777777
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC   77777777
    TT       MMMM  MMMM   PP      PP    AA    AA     CC                 77
    TT       MMMM  MMMM   PP      PP    AA    AA     CC                 77
    TT       MM  MM  MM   PP      PP    AA    AA     CC                 77
    TT       MM  MM  MM   PP      PP    AA    AA     CC                 77
    TT       MM      MM   PPPPPPPP     AA    AA     CC               77
    TT       MM      MM   PPPPPPPP     AA    AA     CC               77
    TT       MM      MM   PP           AAAAAAAAAA     CC             77
    TT       MM      MM   PP           AAAAAAAAAA     CC             77
    TT       MM      MM   PP           AA    AA     CC           77              ....
    TT       MM      MM   PP           AA    AA     CC           77              ....
    TT       MM      MM   PP           AA    AA       CCCCCCCC   77              ....
    TT       MM      MM   PP           AA    AA       CCCCCCCC   77              ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

TMPAC7 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
$TERR	= ******	GX	5-83	5-86	5-87
$TWARN	= ******	GX	5-78		
$VALUE	= ******	GX	7-161	11-311	11-312
.CXLBR	= ******	GX	9-240	*9-243	
.LIBR	= ******	GX	11-300		

TMPAC7 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 3

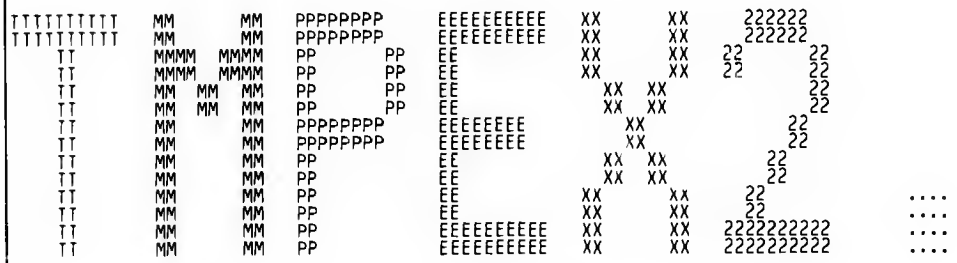
```
                                5$:
                                        .ENDC
                                ;
000040  016700  000000G                  MOV     $LBIAS,R0       ; GET DESCRIPTOR BIAS
000044  017767  000000G 000002'          MOV     @NTLPT,NLBLK    ; SAVE NETLDR DATA BLOCK BIAS
                                ;
000052                                   SWSTK$  20$             ;* ENTER KERNEL MODE
                                ;
000056                                   CALL    $MAPX           ;* MAP TO THE DESCRIPTOR
000062  026700  000002'                  CMP     NLBLK,R0        ;* IS DESCRIPTOR WITHIN NETLDR DATA BLOCK?
                                ;
000066  001402                           BEQ     10$             ;* IF EQ, YES
000070  005262  000002                   INC     2(R2)           ;* TALLY DESCRIPTORS/BLOCK
000074  016702  000000G          10$:    MOV     $LBADR,R2       ;* GET DESCRIPTOR ADDRESS
000100  016722  000000G                  MOV     $FNR50,(R2)+    ;* COPY RAD50 NAME
000104  016722  000002G                  MOV     $FNR50+2,(R2)+  ;*      ...
000110  016722  000004G                  MOV     $FNR50+4,(R2)+  ;*      ...
000114  016722  000000'                  MOV     BFADR,(R2)+     ;* LIBRARY ADDRESS
000120  016722  000000G                  MOV     $BFALL,(R2)+    ;* AND LIBRARY SIZE
000124  005022                           CLR     (R2)+           ;* ZERO THE USE COUNT
                                ;
000126  016700  000000G                  MOV     $BFBAS,R0       ;* GET BASE ADDRESS
000132  006000                           ROR     R0              ;* CONVERT TO AN APR NUMBER
000134                                   ASR$    4,R0            ;*      ...
000144  010042                           MOV     R0,-(R2)        ;* STORE IT
                                ;
000146                          20$:     RETURN                  ;*
                                ;
                                ;
                                ; ERROR CONDITION
                                ;
000150  116777  000000G 000000G 101$:    MOVB    $IOSB,@$CLIOS   ; COPY I/O ERROR CODE
000156                                   EMSG$   Y5              ; LIBRARY FILE READ ERROR
000164  016700  000000'                  MOV     BFADR,R0        ; RETURN WITH LIBRARY ADDRESS IN R0
000170                                   RETURN                  ; AND C-BIT SET
                                ;
                                ;
                000001                   .END
```

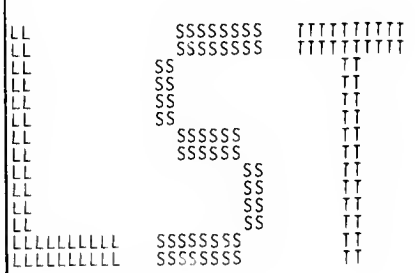
```
                                        .SBTTL  KRBAL - ALLOCATE THE KRB
                                ;+
                                ; *** - KRBAL - ALLOCATE THE KRB
                                ;
                                ; THIS ROUTINE ALLOCATES THE KRB (ONE PER CONTROLLER) AND LINKS IT INTO
                                ; THE SLT FOR ALL LINES ON THIS CONTROLLER.
                                ;
                                ; INPUT:
                                ;       $$PRI = DEVICE PRIORITY
                                ;       $$VECT = VECTOR ADDRESS
                                ;       $$CSR = CSR ADDRESS
                                ;       $SLTA = SLT ADDRESS
                                ;         FOR RSX-11M+, THE FOLLOWING ADDITIONAL INFO:
                                ;       $URM = THE UNIBUS RUN MASK
                                ;
                                ; OUTPUTS:
                                ;       R0-R4 = DESTROYED
                                ;
                                ;-
                                ;
000276                          KRBAL:  .IF DF  R$$11M
                                ;
000276  005000                          CLR     R0              ; KRB ALLOCATION ADDRESS
000300                                  SWSTK$  20$             ;* ENTER KERNEL MODE
000304  012701  000000G                 MOV     #$KRBLN,R1      ;* GET LENGTH OF KRB
000310                                  CALL    @ALOCB          ;* TRY TO ALLOCATE A CORE BLOCK
000314  103402                          BCS     10$             ;* IF CS, COULDN'T
000316  010066  000002                  MOV     R0,2(SP)        ;* RETURN ADDRESS OF CORE BLOCK
000322                          10$:    RETURN                  ;* RETURN TO USER MODE
                                ;
000324  005700                  20$:    TST     R0              ; ALLOCATION SUCCESSFUL ?
000326  001444                          BEQ     101$            ; IF EQ, ALLOCATION FAILURE
                                ;
                                        .IFF    ; DF R$$11M
                                ;
                                        MOV     R3,-(SP)        ; GET SOME FREE REGISTERS
                                        MOV     R4,-(SP)        ;  ...
                                        MOV     #$KRBLN,R3      ; GET LENGTH OF KRB
                                        CLC                     ; ASSUME SUCCESS
                                        .INH0                   ; INHIBIT TASK SWITCHING
                                        CALL    $$RQCB          ;; ALLOCATE MEMORY
                                        ADC     (SP)            ;; SET C-BIT FOR RETURN
                                        .ENB0                   ;; ENABLE TASK SWITCHING
                                        MOV     R4,R0           ; COPY ADDRESS OF BLOCK ALLOCATED
                                        MOV     (SP)+,R4        ; RESTORE REGISTERS
                                        MOV     (SP)+,R3        ;  ...
                                        BCS     101$            ; IF CS, ALLOCATION FAILURE
                                ;
                                        .ENDC   ; DF R$$11M
                                ;
                                        .IF DF  R$$MPL
                                ;
                                        SUB     #K.PRM,R0       ; POINT AT ZERO OFFSET TO KRB
                                ;
                                        .ENDC   ; DF R$$MPL
                                ;
000330  116760  000000G 000000          MOVB    $$PRI,K.PRI(R0) ; STORE THE PRIORITY
```

```
                        .SBTTL  $ALICB - ALLOCATE AN ICB
;+
;*** - $ALICB - ALLOCATE AN ICB
;
;THIS ROUTINE SHOULD BE CALLED TO ALLOCATE AS MANY INTERRUPT
;CONTROL BLOCKS AS ARE NEEDED TO HANDLE THE CURRENT CONTROLLER.
;THE ICB'S ARE THEN INSERTED INTO THE ICB CHAIN STARTING FROM
;K.PRM OF THE KRB.
;
;                       SINGLE PROCESSOR, NON-I&D SPACE
;AN ICB IS ALLOCATED FOR EVERY CONTROLLER IN THE SYSTEM. FOR
;EXAMPLE, A SYSTEM WITH TWO DL'S HAS FOUR ICB'S AND FOUR INTERRUPT
;VECTORS (AN INPUT VECTOR AND AN OUTPUT VECTOR FOR EACH DL).
;EACH INPUT VECTOR POINTS TO IT'S OWN ICB, AND EACH OUTPUT
;VECTOR POINTS TO IT'S OWN ICB. THE ICB'S ARE GROUPED IN BLOCKS,
;ONE INPUT ICB AND ONE OUTPUT ICB PER BLOCK. EACH BLOCK OF ICB'S
;ARE ASSOCIATED WITH ONE CONTROLLER, AND THERE MAY BE UP TO THREE
;ICB'S PER BLOCK (ONE FOR MODEM CONTROL).
;
;THE ICB BLOCKS ARE ALLOCATED OUT OF THE EXECUTIVE DYNAMIC MEMORY POOL.
;
;ICB BLOCKS ARE LINKED TOGETHER SO THEY CAN BE FOUND BY VECTOR CREATION
;(INITL, SAV, AND CON) AND DRIVER UNLOADING (NTL).  THE
;FIRST ICB OF AN ICB CHAIN IS POINTED TO BY K.PRM OF THE APPROPRIATE
;KRB.  A POINTER TO AN ICB IS THE VIRTUAL ADDRESS/2 OF THE FIRST
;WORD OF THE ICB.  A VECTOR THAT POINTS TO AN ICB CONTAINS THE
;VIRTUAL ADDRESS OF THE SECOND WORD (THE MOV BIAS,-(SP) OF THE ICB.
;
;                       MULTIPROCESSOR, NON-I&D SPACE
;BASICALLY THE SAME AS SINGLE PROCESSOR, NON-I&D SPACE SYSTEMS.
;
;                       SINGLE PROCESSOR, I&D SPACE
;BASICALLY THE SAME AS SINGLE PROCESSOR, NON-I&D SPACE SYSTEMS, EXCEPT
;THAT THE ICB IS ALLOCATED FROM A SPECIAL POOL.  THIS MUST BE DONE
;BECAUSE THE EXECUTIVE POOL IS MAPPED AS DATA SPACE, AND THE ICB
;CONTAINS INSTRUCTIONS . THE ICB POOL IS IN THE MODULE LOWCR.
;THE USE OF THAT POOL IS TRANSPARENT, BECAUSE THE NAME OF ITS LISTHEAD
;IS THE SAME WHETHER OR NOT THE POOL IS IN D SPACE OR I&D SPACE.
;
;
;                       MULTIPROCESSOR, I&D SPACE
;EACH PROCESSOR HAS ITS OWN LOW MEMORY CONTEXT.  IT INCLUDES NOT
;ONLY RSX-11M INFORMATION (FOR EXAMPLE, WHICH TASK IS THE ACTIVE
;TASK ON THE PROCESSOR) BUT THE INTERRUPT VECTORS AND ASSOCIATED
;ICB'S FOR THE PERIPHERALS CONNECTED (EVEN THROUGH A BUS SWITCH)
;TO THE PROCESSOR'S UNIBUS.  THEREFORE EVERY CPU NEEDS AN ICB FOR
;EVERY VECTOR TYPE OF EVERY CONTROLLER CONNECTED TO THE CPU'S UNIBUS.
;FOR EXAMPLE, TWO DL'S (EACH ONE ON A DIFFERENT CPU) REQUIRE FOUR ICB'S.
;AN ICB FOR EACH INPUT VECTOR ON EACH CPU, AND AN ICB FOR EACH OUTPUT
;VECTOR ON EACH CPU.
;
;TO AVOID HAVING TO CREATE ICB'S ON THE FLY (FOR EXAMPLE, WHEN A
;BUS SWITCH IS SWITCHED), ALL ICB'S THAT ARE POSSIBLY NEEDED ARE
;CREATED WHEN THE DRIVER IS LOADED.
;
;THE ICB'S ARE LINKED TOGETHER BY A 16-BIT POINTER IN EACH ICB.
;THE LISTHEAD IS THE WORD L.ICB OF THE APPROPRIATE CTB.  THE
```

```
                        .TITLE  TMPERR - NTL TEMPLATE ERROR PROCESSING
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - "NTLTMP" AND "TMPSTB" ERROR MESSAGES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
**FILE**ID**TMPEX2
```





```
                                        .IFF
                                        TSTB    -5(R5)          ; IS THIS .CORE OR .BIN?
                                        BPL     10$             ; IF PL, .BIN
                                        MOV     -2(R4),R2       ; GET ALLOCATION ADDRESS
                                        ASR     R3              ; CONVERT TO A WORD COUNT
                                        ADD     #77,R3          ;       ...
                                        BIC     #77,R3          ;       ...
                                        ASR     R3              ;       ...
                                        .ENDC
001020  005022                  5$:     CLR     (R2)+           ;* ZERO THE ALLOCATION
001022                                  SOB     R3,5$           ;*      ...
                                        .IF DF  R$$11D ! I$$AS
                                        CALL    @(SP)+          ;* RESTORE PREVIOUS MAPPING
                                        .ENDC
001026                          10$:    RETURN
```

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 3

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES							
SLNB	000444	R	6-113	#9-339						
SLNW	000470	R	6-114	#9-350						
STNB	000536	R	6-115	#9-377						
STNW	000574	R	6-116	#9-390						
S.COST	000001		#5-64							
S.FLG	000000		#5-64							
S.LEN	000004		#5-64							
S.NMST	000002		#5-64							
S.OWNR	000003		#5-64							
TABLE	000000	R	#6-89	7-183	13-583	13-608	15-687			
TIME	001612	R	6-123	#15-704						
UNB	000636	R	6-117	#9-405						
UNW	000660	R	6-118	#9-415						
WORD	000702	R	6-119	#9-425						
X2CH01	001774	R	6-137	#17-798						
X2CH02	002022	R	6-138	#17-808						
X2CH03	002050	R	6-139	#17-818						
X2CH04	002076	R	6-140	#17-828						
X2CH05	002124	R	6-141	#17-838						
X3CH01	001672	R	6-126	#16-741						
X3CH02	001700	R	6-127	#16-746						
X3CH03	001706	R	6-128	#16-751						
X3CH04	001714	R	6-129	#16-756						
X3CH05	001722	R	6-130	#16-761						
X3CH06	001730	R	6-131	#16-766						
X3CH07	001736	R	6-132	#16-771						
X3CH08	001744	R	6-133	#16-776						
X3CH09	001752	R	6-134	#16-781						
X3CH10	001760	R	6-135	#16-786						
X3CH11	001766	R	6-136	#16-791						
ZTIME	= ******	GX	15-704							
$ADDR	= ******	GX	7-176	9-316	13-610					
$BIAS	= ******	GX	12-555							
$BIN	= ******	GX	7-175							
$CTIM	= ******	GX	9-249							
$FLAGS	= ******	GX	8-195	8-205	9-255	9-257	9-280	9-282	9-325	17-799	17-809
			17-819	17-829	17-839					
$LDA	= ******	GX	14-663	*14-667						
$MAPX	= ******	GX	10-456	11-533	14-651					
$MXSIZ	= ******	GX	15-697							
$PTR	= ******	GX	7-177							
$SLN	= ******	GX	9-344	9-356						
$SLTSV	= ******	GX	9-266							
$TMEX2	000000	RG	#7-172							
$UBIAS	= ******	GX	10-455	11-532	14-653					
$UDA	= ******	GX	10-460	*10-475	11-537	*11-540				
$X2HLD	= ******	GX	17-843							
$X2MBS	= ******	GX	17-813							
$X2MRT	= ******	GX	17-823							
$X2MWS	= ******	GX	17-803							
$X2RET	= ******	GX	17-833							
$X3CAL	= ******	GX	16-761							

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 4

MACRO DEFINTIONS

```
                                .SBTTL  MACRO DEFINTIONS
;
; LIBRARY MACROS
;
                                .MCALL  EMSG$,EMSG$R,NTLER$,ASR$
                                .IF DF R$$MPL
                                .MCALL  STAF$,RMAF$S,DIR$
                                .ENDC
```


LOCAL DATA

F 2

TMPACO CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES							
ASR$	#5-60	13-226						
CALL	8-123	9-143	10-160	11-177	12-194	13-212	14-277	14-285
CALLR	9-147	10-164	11-181	12-198				
EMSG$	#5-60							
EMSG$R	#5-60	13-261	13-262	13-263				
NTLER$	#5-60	7-100	7-101	7-102				
RETURN	8-129	13-256	14-287					

G 2

FILEID**TMPAC2

TMPAC2 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:39 Page 12-1
Symbol table

```
A$$CHK= 000000          F1.MXT= ****** GX       N$$LDV= 000001          $ELINE= ****** GX       $NALL = ****** GX
A$$CPS= 000000          F2.FIL= ****** GX       N$$MCP= 000001          $ERRKX  000316R     002 $NLIB = ****** GX
A$$PRI= 000000          G$$TPP= 000000          N$$MLL= 000001          $ERRKY  000362R     002 $OFF  = ****** GX
A$$TRP= 000000          G$$TSS= 000000          N$$MOV= 000010          $ERR17  000004R     002 $PRV.N= ****** GX
CTLERR  000624R         G$$TTK= 000000          N$$NCT= 000001          $ERR22  000204R     002 $REP.C= ****** GX
C$$CKP= 000000          G$$WRD= 000000          N$$PEM= 000001          $ERR41  000042R     002 $RQ1  = ****** GX
C$$ORE= 000400          I$$RAR= 000000          P$$P45= 000000          $ERR42  000104R     002 $RQ2  = ****** GX
C$$RSH= 177564          I$$RDN= 000000          P$$WRD= 000000          $ERR43  000146R     002 $RQ4  = ****** GX
D$$BUG= 177514          K$$CNT= 177546          Q$$OPT= 000010          $ERR81  000242R     002 $SECSR  001020RG
D$$ISK= 000000          K$$CSR= 177546          R$$DER= 000000          $ERR85  000270R     002 $SIZE = ****** GX
D$$L11= 000001          K$$LDC= 000000          R$$K11= 000001          $EVEN = ****** GX       $TERR = ****** GX
D$$YNC= 000000          K$$TPS= 000074          R$$SND= 000000          $FILE   000000RG        $TMFIX= ****** GX
D$$YNM= 000000          LD$LP = 000000          R$$11M= 000000          $FILEN= ****** GX       $VALUE= ****** GX
E$$XPR= 000000          L$$ASG= 000000          S$$WRG= 000000          $FILEX  000050RG        $VFDDM= ****** GX
FL.CHA= ****** GX       L$$DRV= 000000          S$$YSZ= 007600          $FLAGS= ****** GX       $VFDLC= ****** GX
FL.LMC= ****** GX       L$$P11= 000001          TMPEXT  000000R     002 $FLAG1= ****** GX       $VFMUX  001070RG
FL.MUX= ****** GX       L$$11R= 000000          T$$KMG= 000000          $FLAG2= ****** GX       $VFSPC= ****** GX
FMT2    000442R     002 MAX14   000336R         T$$MIN= 000000          $LFILE  000436RG        $$MTP = ****** GX
FMT3    000461R     002 M$$CRB= 000124          V$$CTR= 001000          $LFLD = ****** GX       $$MUX = ****** GX
FM.2  = 000000          M$$CRX= 000000          X$$DBT= 000000          $MPLD = ****** GX       .EOF  = ****** GX
FM.3  = 000000          M$$FCS= 000000          $ALL16= ****** GX       $MPTB   000370RG        .FILE = ****** GX
F$$LVL= 000001          M$$MGE= 000000          $AZFS = ****** GX       $MXADD= ****** GX       .LFILE= ****** GX
F1.LFI= ****** GX       M$$NET= 000000          $CFILE  000704RG        $MXLEN= ****** GX       .MPTAB= ****** GX
F1.LFL= ****** GX       M$$OVR= 000000          $CLIOS= ****** GX       $MXPTB  000064RG        .MXPTB= ****** GX
F1.MPL= ****** GX       N$$ACC= 000001          $CLQUE= ****** GX       $MXSIZ= ****** GX       .MXTAB= ****** GX
F1.MPT= ****** GX       N$$BUF= 000001          $EFNLL= ****** GX       $MXTAB  000632RG        .SECSR= ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001110    001   (RW,I,LCL,REL,CON)
DATA    000506    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10250 Words  ( 41 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:16.01
SY:TMPAC2.V2,[132,134]TMPAC2/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC2
```


TMPAC2 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

```
**FILE**ID**TMPAC4
```

```
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC   44    44
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC   44    44
    TT       MMMM  MMMM   PP      PP    AA      AA   CC         44    44
    TT       MMMM  MMMM   PP      PP    AA      AA   CC         44    44
    TT       MM  MM  MM   PP      PP    AA      AA   CC         44    44
    TT       MM  MM  MM   PP      PP    AA      AA   CC         44    44
    TT       MM      MM   PPPPPPPP      AA      AA   CC         4444444 44
    TT       MM      MM   PPPPPPPP      AA      AA   CC         4444444444
    TT       MM      MM   PP            AAAAAAAAAA   CC               44
    TT       MM      MM   PP            AAAAAAAAAA   CC               44
    TT       MM      MM   PP            AA      AA   CC               44        ....
    TT       MM      MM   PP            AA      AA   CC               44        ....
    TT       MM      MM   PP            AA      AA     CCCCCCCC       44        ....
    TT       MM      MM   PP            AA      AA     CCCCCCCC       44        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

```
SYMBOL  VALUE           REFERENCES

$CLIOS  = ******  GX     6-183
$CORE     000306 RG     #8-242
$CORE1    000236 RG     #7-205
$CUR.N  = ******  GX     5-90
$CVAL   = ******  GX     *7-215      *8-255      8-256       8-281       8-282
$EFNLL  = ******  GX     5-86        5-90
$ELINE  = ******  GX     5-84        5-91
$ERR26    000000 R      #5-84       6-192       8-295
$ERR76    000044 R      #5-86       6-141
$ERR77    000100 R      #5-90       6-184
$ERR78    000136 R      #5-91       10-357
$FLAGS  = ******  GX     6-127       8-253       9-313
$FLAG1  = ******  GX     6-147       8-262
$FLAG2  = ******  GX     6-124
$IOSB   = ******  GX     6-183
$LBN    = ******  GX     *6-178      *6-179
$LLEN   = ******  GX     *6-180
$NALL   = ******  GX     *6-155      *8-270      *9-325      10-346
$NLIB   = ******  GX     10-348
$READX  = ******  GX     6-181
$REP.C  = ******  GX     5-84        5-91        11-374
$RQ5    = ******  GX     9-310
$RQ7    = ******  GX     8-246
$SCOM     000452 RG     #9-308
$SIZE   = ******  GX     *6-171      *8-285      *9-332
$TERR   = ******  GX     5-84        5-90        5-91
$TWARN  = ******  GX     5-86
$VALUE  = ******  GX     6-132       6-163       6-164       7-209       *7-216      8-277       8-278       *9-315      9-316
                        *9-322      9-330       9-331
$VFLV1  = ******  GX     8-243       9-309
.BASEB  = 140000        #5-64       5-76        7-216
.CORE   = ******  GX     6-157       8-271
.SCOM   = ******  GX     9-326
```


TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 3

MACRO NAME	REFERENCES								
ASL$	#5-59	8-231							
CALL	6-138	6-141	6-151	6-153	6-155	7-172	7-173	7-184	8-202
EMSG$	#5-59								
EMSG$R	#5-59	7-183	7-185	8-245	8-246	8-247	8-248		
NHWDF$	#5-59	5-64							
NTLER$	#5-59	5-77	5-78	5-79	5-80	5-82	5-86		
RETURN	6-157	7-178	8-239						

```
                                        .SBTTL  $LINKS - .LINKS ACTION ROUTINE
                                ;+
                                ; $LINKS - ".LINKS" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R2=DESTROYED
                                ;-
000574                          $LINKS::
000574                                  CALL    $VFLLC          ; MAKE SURE IT'S AN LLC
000600                                  CALL    $RQ1            ; REQUEST BINARY BUFFER ACCESS
000604  103405                         BCS     10$             ; IF CS, NO ACCESS
000606  112712  000000G                MOVB    #.LINKS,(R2)    ; STORE FUNCTION CODE
000612  062767  000002  000000G        ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
000620                          10$:    RETURN
```

```
                        .TITLE  TMPAC7 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 7
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

TMPAC7 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	6-127	6-129	6-133	6-137	6-140	6-142	7-170	7-171	7-178	7-182
	7-184	7-192	7-195	7-198	7-199	8-223	9-238	9-239	10-269	10-272
	10-284	11-318	11-320	12-351	12-361					
EMSG$	#5-59	7-170	10-284							
EMSG$R	#5-59	7-193	7-196	8-224						
NTLER$	#5-59	5-78	5-83	5-86	5-87					
RETURN	6-143	7-186	7-201	8-218	9-251	10-277	10-285	11-324	11-339	12-362
	13-394									
SWSTK$	9-238	10-269	11-318							

FILEID**TMPAC8

TMPAC9 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:41 Page 6-2
Symbol table

```
A$$CHK= 000000          F$$LVL= 000001          M$$CRB= 000124          P$$P45= 000000          $BFXFR= ****** GX
A$$CPS= 000000          G$$TPP= 000000          M$$CRX= 000000          P$$WRD= 000000          $CLDEQ= ****** GX
A$$PRI= 000000          G$$TSS= 000000          M$$FCS= 000000          Q$$OPT= 000010          $CLIOS= ****** GX
A$$TRP= 000000          G$$TTK= 000000          M$$MGE= 000000          R$$DER= 000000          $CUR.N= ****** GX
BFADR   000000R     002 G$$WRD= 000000          M$$NET= 000000          R$$K11= 000001          $EFNLL= ****** GX
C$$CKP= 000000          I$$RAR= 000000          M$$OVR= 000000          R$$SND= 000000          $ERRY5  000004R     002
C$$ORE= 000400          I$$RDN= 000000          NLBLK   000002R     002 R$$11M= 000000          $FNR50= ****** GX
C$$RSH= 177564          K$$CNT= 177546          NTLPT = ****** GX       S$$WRG= 000000          $IOSB = ****** GX
D$$BUG= 177514          K$$CSR= 177546          N$$ACC= 000001          S$$YSZ= 007600          $LBADR= ****** GX
D$$ISK= 000000          K$$LDC= 000000          N$$BUF= 000001          T$$KMG= 000000          $LBIAS= ****** GX
D$$L11= 000001          K$$TPS= 000074          N$$LDV= 000001          T$$MIN= 000000          $LBN  = ****** GX
D$$YNC= 000000          LD$LP = 000000          N$$MCP= 000001          V$$CTR= 001000          $LBXFR  000000RG
D$$YNM= 000000          L$$ASG= 000000          N$$MLL= 000001          X$$DBT= 000000          $LLEN = ****** GX
E$$XPR= 000000          L$$DRV= 000000          N$$MOV= 000010          $BFALL= ****** GX       $MAPX = ****** GX
FMT6  = ****** GX       L$$P11= 000001          N$$NCT= 000001          $BFBAS= ****** GX       $READX= ****** GX
FM.6  = 000000          L$$11R= 000000          N$$PEM= 000001          $BFLBN= ****** GX       $TERR = ****** GX

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000172    001   (RW,I,LCL,REL,CON)
DATA    000042    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10159 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:07.59
SY:TMPAC9.V2,[132,134]TMPAC9/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC9
```


TMPAC9 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

KRBAL - ALLOCATE THE KRB

```
363 000336  016701  000000G                 MOV     $$VECT,R1       ; GET THE VECTOR ADDRESS
364 000342  006201                          ASR     R1              ; DIVIDED BY 4
365 000344  006201                          ASR     R1              ;
366 000346  110160  000001                  MOVB    R1,K.VCT(R0)    ; STORE THE VECTOR/4
367 000352  016760  000000G 000002          MOV     $$CSR,K.CSR(R0) ; STORE THE CSR ADDRESS
368                                 ;
369                                 ; STORE THE KRB ADDRESS IN ALL SLT'S ON THIS CONTROLLER
370                                 ;
371 000360  016701  000000G                 MOV     $SLTA,R1        ; GET THE SLT ADDRESS
372 000364  116102  000002                  MOVB    L.DDM(R1),R2    ; SAVE THE DDM PDV INDEX FOR THIS LINE
373 000370  116103  000012                  MOVB    L.CTL(R1),R3    ; SAVE THE CONTROLLER NUMBER FOR THIS LINE
374
375                                         .IF DF  R$$MPL
376
377                                         MOVB    R3,K.CON(R0)    ; STORE THE CONTROLLER NUMBER IN THE KRB
378                                         ASLB    K.CON(R0)       ; GET CONTROLLER NUMBER*2
379
380                                         .IFTF   ; DF R$$MPL
381
382 000374  017701  000000G                 MOV     @SLTMA,R1       ; GET THE START OF THE SLT TABLE
383 000400  017746  000000G                 MOV     @SLTNM,-(SP)    ; SAVE THE NUMBER OF SLT ENTRIES ON STACK
384 000404  001420                          BEQ     111$            ; IF EQ, NO ENTRIES .. LOGIC ERROR
385
386 000406  012104                  30$:    MOV     (R1)+,R4        ; GET NEXT SLT ADDRESS
387 000410  120264  000002                  CMPB    R2,L.DDM(R4)    ; IS THIS THE SAME DEVICE TYPE ?
388 000414  001005                          BNE     35$             ; IF NE, NO .. KEEP LOOKING
389 000416  120364  000012                  CMPB    R3,L.CTL(R4)    ; IS THIS THE SAME CONTROLLER ?
390 000422  001002                          BNE     35$             ; IF NE, NO .. KEEP LOOKING
391                                 ;
392                                 ; FOUND ONE
393                                 ;
394 000424  010064  000016                  MOV     R0,L.KRBA(R4)   ; STORE THE KRB ADDRESS FOR THIS LINE
395 000430  005316                  35$:    DEC     (SP)            ; REPEAT <$SLTNM> TIMES
396 000432  003365                          BGT     30$             ;
397 000434  005726                  37$:    TST     (SP)+           ; PURGE THE STACK
398
399
400
401                                         CLRB    K.IOC(R0)       ; INITIALIZE THE I/O COUNT
402                                         MOV     #KS.OFL,K.STS(R0) ; INITIALIZE THE STATUS WORD, OFFLINE
403                                         CLRB    K.HPU(R0)       ; CLEAR THE HIGHEST UNIT NUMBER
404                                         CLR     K.OWN(R0)       ; NO OWNER TASK
405                                         MOV     $URM,K.URM(R0)  ; STORE THE UNIBUS RUN MASK
406                                         CLR     K.CRQ(R0)       ; INITIALIZE THE CONTROLLER REQUEST QUEUE
407                                         MOV     R0,R1           ;       ...
408                                         ADD     #K.CRQ,R1       ;       ...
409                                         MOV     R1,K.CRQ+2(R0)  ;       ...
410                                 ;
411                                 ; INSERT KRB ADDRESS INTO CTB
412                                 ;
413                                         MOVB    K.CON(R0),R1    ; GET THE CONTROLLER NUMBER*2
414                                         ADD     $CTBAD,R1       ; CALCULATE THE START OF THE CTB + OFFSET
415                                         MOV     R0,L.KRB(R1)    ; SET KRB ADDRESS IN CTB
416                                         CALL    $HRCIN          ; RE-INITIALIZE HRC...
417                                         BCC     40$             ; IF CC, ALL OKAY
418                                         EMSG$   9Y              ; ELSE, WARN SOMEBODY
419                                         BIC     #F1.ERR,$FLAG1  ; THIS IS ONLY A WARNING
```

```
                        ;
                        ; POINTERS NEED TO BE 18 BITS, BECAUSE THE POOLS THAT CONTAIN THE
                        ; ICB'S ARE IN THE CPU PARTITIONS, WHICH  CAN BE ANYWHERE IN THE 124K
                        ; OF A SAVED SYSTEM IMAGE.  TO GET AROUND THIS, THE POINTERS ARE ENCODED
                        ; VALUES.  THE UPPER FOUR BITS ARE THE NUMBER OF THE CPU TO WHICH THE
                        ; ICB BELONGS.  USING THAT AS AN INDEX INTO THE TABLE AT $K6TAB GIVES
                        ; THE BLOCK NUMBER (32. WORDS PER BLOCK) OF THE BASE OF THE CPU DEPENDENT
                        ; LOW MEMORY CONTEXT.  THE LOWER 12 BITS ARE THE VIRTUAL ADDRESS/2 OF THE
                        ; ICB WITHIN THAT LOW MEMORY CONTEXT.  NOTE THAT THIS LIMITS THE
                        ; SIZE OF A LOW MEMORY CONTEXT TO A MAXIMUM OF 4K.
                        ;
                        ; INPUTS:
                        ;       $SLTA = THE SLT ADDRESS FOR THIS LINE
                        ;       CALLED AT SYSTEM STATE
                        ;
                        ; OUTPUTS:
                        ;       C-BIT = SUCCESS/FAILURE
                        ;       IF SUCCESS:
                        ;               ALL ICBS WERE ALLOCATED AND LINKED TOGETHER
                        ;               R0 = ZERO IF THIS IS A MULTI-PROCESSOR STSTEM
                        ;               R0 = THE ICB ADDRESS IF THIS IS NOT A MULTI-PROCESSOR SYSTEM
                        ;       R1-R4 = DESTROYED
                        ;-
                        ;
                                .IF DF  R$$MPL
                        $ALICB::
                                TSTB    @NCPU           ;* MORE THAN 1 CPU ?
                                BNE     20$             ;* IF NE, YES
                                MOV     ICAVL,R0        ;* GET ADDRESS-2 OF ICB POOL LISTHEAD
                                SUB     #2,R0           ;* ...
                                MOV     $ICBSZ,R1       ;* AND LENGTH OF AN ICB
                                CALL    @ALOC1          ;* ALLOCATE THE ICB
                                BCS     10$             ;* IF CS, ALLOCATION FAILURE
                                MOVB    $NVECT,(R0)+    ;* STORE THE NUMBER OF ICB'S
                                MOVB    $ICBLN,(R0)+    ;* AND THE LENGTH OF EACH ICB
                                MOV     $SLTA,R1        ;* GET THE SLT ADDRESS
                                MOV     L.KRBA(R1),R1   ;* GET THE KRB ADDRESS
                                MOV     R0,R2           ;* COPY THE ICB ADDRESS
                                ASR     R2              ;* DIVIDE THE ADDRESS BY 2
                                MOV     R2,K.PRM(R1)    ;* STORE THE ADDRESS IN THE KRB
                                CLR     (R0)            ;* CLEAR THE LINK POINTER
                        10$:    RETURN                  ;* RETURN TO CALLER
                        ;
                        ; IF THE CONTROLLER IS ON A SWITCHED BUS, DETERMINE WHICH
                        ; PROCESSORS THE BUS CAN BE SWITCHED TO.
                        ;
                        20$:    MOV     @CPMSK,R2       ;* GET CPU RUN MASK
                                MOV     @KSAR5,-(SP)    ;* SAVE CURRENT MAPPING
                                MOV     $SLTA,R4        ;* GET THE SLT ADDRESS
                                MOV     L.KRBA(R4),R4   ;* GET THE KRB ADDRESS
                                MOV     K.URM(R4),R0    ;* GET UNIBUS RUN MASK
                                ADD     #K.PRM,R4       ;* POINT R4 AT START OF ICB CHAIN
                                CLR     (R4)            ;* ZERO LINK TO NEXT ICB
                                BIT     R2,R0           ;* IS CONTROLLER ON A SWITCHED BUS?
                                BEQ     100$            ;* IF EQ NO
                                MOV     DEVHD,R1        ;* POINT AT DEVICE LIST HEAD
                        30$:    MOV     (R1),R1         ;* GET NEXT DEVICE IN LIST
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  EMSG$R,NTLER$
                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D


                                ; "NTLTMP" ERROR MESSAGES (PLACED HERE FOR OVERLAY PURPOSES)
                                ;
                                        .ENABL  LC

000000                                  NTLER$  <:>,9,2,$TERR,$REP.C,$ELINE,<Syntax Error>
000026                                  NTLER$  <:>,10,3,$TERR,$PRV.N,,<Read Failure (-***.)>
000064                                  NTLER$  <:>,11,3,$TWARN,$REP.P,,<No End Statement>
000116                                  NTLER$  <:>,12,3,$TERR,$REP.P,$ELINE,<Read Overflow>


                                ; "TMPSTB" ERROR MESSAGES (PLACED HERE FOR OVERLAY PURPOSES)
                                ;
000144                                  NTLER$  <:>,1,2,$TERR,$REP.C,$ELINE,<Undefined Operator>
000200                                  NTLER$  <:>,3,2,$TERR,$REP.C,$ELINE,<No Operands Allowed>
000234                                  NTLER$  <:>,4,2,$TERR,$REP.C,$ELINE,<Only One Operand Allowed>
000276                                  NTLER$  <:>,5,2,$TERR,$REP.C,$ELINE,<Only Two Operands Allowed>
000340                                  NTLER$  <:>,6,2,$TERR,$REP.C,$ELINE,<Operand Missing>
000370                                  NTLER$  <:>,7,2,$TERR,$REP.C,$ELINE,<Illegal Operand>

                                        .DSABL  LC


                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000420  052  040  124  FMT2:    .ASCIZ  "* Template -- "
000437  052  040  124  FMT3:    .ASCIZ  "* Template File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

000000                                  .PSECT
```

Table of contents


```
     5-   55    MACRO DEFINITIONS
     6-   78    LOCAL DATA
     7-  155    $TMEX2 - EXPAND BINARY INFO
     8-  187    ADDRB,ADDRW,BLKB,BLKW
    16-  736    X3CH
    17-  794    X2CH
```

```
                                ;
                                ; POINTER TO ALLOCATED SCOM
                                ;
                                        .ENABL  LSB
001030  112524                SCOM:   MOVB    (R5)+,(R4)+     ; COPY ADDRESS
001032  112524                        MOVB    (R5)+,(R4)+     ;       ...
001034  112546                        MOVB    (R5)+,-(SP)     ; ASSEMBLE COUNT
001036  112566  000001                MOVB    (R5)+,1(SP)     ;       ...
001042  012603                        MOV     (SP)+,R3        ;       ...
001044  001425                        BEQ     10$             ; IF ZERO, NO UNLOAD DATA TO STORE
001046  016400  177776                MOV     -2(R4),R0       ; GET SCOM ADDRESS
001052  010346                        MOV     R3,-(SP)        ; SAVE R3
001054  006203                        ASR     R3              ; CONVERT TO A WORD COUNT
001056  005020                5$:     CLR     (R0)+           ; ZERO THE ALLOCATED SCOM
001060                                SOB     R3,5$           ;       ...
001064  012603                        MOV     (SP)+,R3        ; RESTORE R3

001066                          SAVALC: .IF DF  M$$MGE

                                        .IF DF  R$$11M

001066                                  SWSTK$  10$             ;* ENTER KERNEL MODE

                                        .ENDC

001072  016700  000000G                 MOV     $UBIAS,R0       ;* GET UNLOAD BLOCK BIAS
001076                                  CALL    $MAPX           ;* MAP TO UNLOAD BLOCK

                                        .ENDC

001102  016702  000000G                 MOV     $UDA,R2         ;* GET UNLOAD DATA ADDRESS
001106  016422  177776                  MOV     -2(R4),(R2)+    ;* STORE ALLOCATION ADDRESS
001112  010322                          MOV     R3,(R2)+        ;* STORE COUNT
001114  010267  000000G                 MOV     R2,$UDA         ;* SET NEW UNLOAD DATA ADDRESS

                                        .IF DF  R$$11D ! I$$AS
                                        CALL    @(SP)+          ;* CO-ROUTINE CALL TO RESTORE MAPPING
                                        .ENDC

001120                          10$:    RETURN
                                        .DSABL  LSB
```

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 4

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
$X3CLR	= ******	GX	16-766		
$X3DBS	= ******	GX	16-741		
$X3DWS	= ******	GX	16-751		
$X3MBS	= ******	GX	16-746		
$X3MCL	= ******	GX	16-781		
$X3MRE	= ******	GX	16-786		
$X3MRS	= ******	GX	16-791		
$X3MWS	= ******	GX	16-756		
$X3RES	= ******	GX	16-771		
$X3RST	= ******	GX	16-776		
$$APR	= ******	GX	15-717		
$$CSR	= ******	GX	9-242		
$$DCHA	= ******	GX	9-269	9-270	
$$DPR	= ******	GX	15-726	15-732	
$$LSAD	= ******	GX	8-197	8-208	
$$MTP	= ******	GX	9-379	9-392	
$$PCHA	= ******	GX	9-289		
$$PRI	= ******	GX	12-568		
$$SCSR	= ******	GX	9-331		
$$SNUM	= ******	GX	9-382	9-396	
.EOF	= ******	GX	13-580	13-605	15-684

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 5

```
                                        .SBTTL  LOCAL DATA

000000                                  .PSECT  DATA,D

                                ;
                                ; NONSENSE INTERRUPT TABLE (RSX11M)
                                ;
                                        .IF DF  R$$11M
                                NSITAB: .WORD   NS0
000002  000000G                         .WORD   NS1
000004  000000G                         .WORD   NS2
000006  000000G                         .WORD   NS3
000010  000000G                         .WORD   NS4
000012  000000G                         .WORD   NS5
000014  000000G                         .WORD   NS6
000016  000000G                         .WORD   NS7
                                        .ENDC

                                        .IF DF  R$$MPL

                                SETAF:  STAF$   0

                                        .ENDC
```

```
**FILE**ID**TMPAC2
```

```
TTTTTTTTTT    MM      MM    PPPPPPPP       AAAAAA       CCCCCCCC     222222
TTTTTTTTTT    MM      MM    PPPPPPPP       AAAAAA       CCCCCCCC     222222
    TT        MMMM  MMMM    PP      PP    AA      AA    CC          22      22
    TT        MMMM  MMMM    PP      PP    AA      AA    CC          22      22
    TT        MM  MM  MM    PP      PP    AA      AA    CC                  22
    TT        MM  MM  MM    PP      PP    AA      AA    CC                  22
    TT        MM      MM    PPPPPPPP      AA      AA    CC              22
    TT        MM      MM    PPPPPPPP      AA      AA    CC              22
    TT        MM      MM    PP            AAAAAAAAAA    CC            22
    TT        MM      MM    PP            AAAAAAAAAA    CC            22
    TT        MM      MM    PP            AA      AA    CC          22          ....
    TT        MM      MM    PP            AA      AA    CC          22          ....
    TT        MM      MM    PP            AA      AA      CCCCCCCC  2222222222  ....
    TT        MM      MM    PP            AA      AA      CCCCCCCC  2222222222  ....
```

```
LL              SSSSSSSS    TTTTTTTTTT
LL              SSSSSSSS    TTTTTTTTTT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL            SS                TT
LL              SSSSSS          TT
LL              SSSSSS          TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LL                    SS        TT
LLLLLLLLLL    SSSSSSSS          TT
LLLLLLLLLL    SSSSSSSS          TT
```

TMPAC2 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES							
CTLERR	000624 R	9-273	#9-303	10-336					
FL.CHA	= ****** GX	7-190							
FL.CMC	= ****** GX	11-371							
FL.MUX	= ****** GX	12-394							
FMT2	000442 R	5-82	5-83	5-84	5-85	5-86	5-87	5-88	5-89	#5-98
FMT3	000461 R	5-81	#5-99						
FM.2	= 000000	#5-82	#5-83	#5-84	#5-85	#5-86	#5-87	#5-88	#5-89	
FM.3	= 000000	#5-81							
F1.LFI	= ****** GX	9-265							
F1.LFL	= ****** GX	9-268							
F1.MPL	= ****** GX	9-258							
F1.MPT	= ****** GX	9-255							
F1.MXT	= ****** GX	7-161	10-324						
F2.FIL	= ****** GX	6-121							
MAX14	000336 R	7-169	#8-226						
S$$BAS	= ******	5-81	5-81	5-82	5-82	5-83	5-83	5-84	5-84	5-85
		5-85	5-86	5-86	5-87	5-87	5-88	5-88	5-89	5-89
TMPEXT	000000 R	#5-72	6-125							
$ALL16	= ****** GX	7-195								
$AZFS	= ****** GX	6-126								
$CFILE	000704 RG	#10-332								
$CLIOS	= ****** GX	6-128								
$CLQUE	= ****** GX	6-127								
$EFNLL	= ****** GX	5-81								
$ELINE	= ****** GX	5-82	5-83	5-84	5-85	5-86	5-87	5-88	5-89	
$ERRKX	000316 R	#5-88	7-211							
$ERRKY	000362 R	#5-89	8-237							
$ERR17	000004 R	#5-81	6-139							
$ERR22	000204 R	#5-85	7-210	9-303						
$ERR41	000042 R	#5-82	9-301							
$ERR42	000104 R	#5-83	9-302							
$ERR43	000146 R	#5-84	9-292							
$ERR81	000242 R	#5-86	12-396							
$ERR85	000270 R	#5-87	11-382							
$EVEN	= ****** GX	7-178	9-285	10-333	10-345					
$FILE	000000 RG	#6-120	7-177	9-284	10-344					
$FILEN	= ****** GX	6-124								
$FILEX	000050 RG	#6-132								
$FLAGS	= ****** GX	7-190	11-371	12-394						
$FLAG1	= ****** GX	9-258	9-268							
$FLAG2	= ****** GX	6-121								
$LFILE	000436 RG	#9-264								
$LFLD	= ****** GX	9-271								
$MPLD	= ****** GX	9-261								
$MPTB	000370 RG	#9-253								
$MXADD	= ****** GX	*7-198								
$MXLEN	= ****** GX	*7-188	*10-330							
$MXPTB	000064 RG	#7-158								
$MXSIZ	= ****** GX	*7-194								
$MXTAB	000632 RG	#10-321								
$NALL	= ****** GX	*7-197	8-227							
$NLIB	= ****** GX	8-229								

TMPAC2 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

```
                        .TITLE  TMPAC4 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 4
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

TMPAC4 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 3

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASR$	#5-59	7-214								
CALL	6-141	6-142	6-149	6-153	6-181	6-184	6-185	6-191	8-243	8-246
	8-258	8-264	8-268	9-309	9-310	9-318	9-323	11-374		
EMSG$	#5-59	6-141	6-184							
EMSG$R	#5-59	6-192	8-295	10-357						
NHWDF$	#5-59	5-64								
NTLER$	#5-59	5-84	5-86	5-90	5-91					
RETURN	6-186	7-225	8-290	9-333	10-352	11-383				

FILEID**TMPAC5

FILEID**TMPAC6

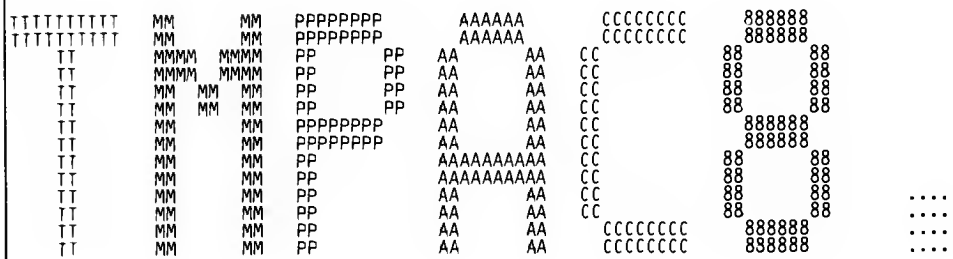
```
TTTTTTTTTT    MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC     666666
TTTTTTTTTT    MM      MM   PPPPPPPP       AAAAAA       CCCCCCCC     666666
    TT        MMMM  MMMM   PP      PP     AA      AA   CC           66
    TT        MMMM  MMMM   PP      PP     AA      AA   CC           66
    TT        MM  MM  MM   PP      PP     AA      AA   CC           66
    TT        MM  MM  MM   PP      PP     AA      AA   CC           66
    TT        MM      MM   PPPPPPPP       AA      AA   CC           66666666
    TT        MM      MM   PPPPPPPP       AA      AA   CC           66666666
    TT        MM      MM   PP             AAAAAAAAAA   CC           66      66
    TT        MM      MM   PP             AAAAAAAAAA   CC           66      66
    TT        MM      MM   PP             AA      AA   CC           66      66       ....
    TT        MM      MM   PP             AA      AA   CC           66      66       ....
    TT        MM      MM   PP             AA      AA     CCCCCCCC     666666         ....
    TT        MM      MM   PP             AA      AA     CCCCCCCC     666666         ....


LL              SSSSSSSS   TTTTTTTTTT
LL              SSSSSSSS   TTTTTTTTTT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL            SS               TT
LL              SSSSSS         TT
LL              SSSSSS         TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LL                    SS       TT
LLLLLLLLLL    SSSSSSSS         TT
LLLLLLLLLL    SSSSSSSS         TT
```

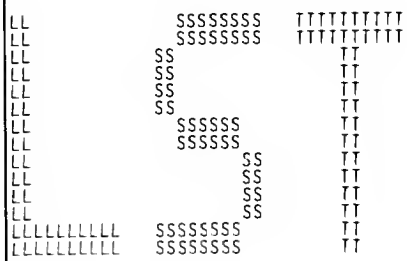

TMPAC6 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:40

```
                                        .SBTTL  $UMRON - .UMR ON ACTION ROUTINE
                                ;+
                                ; $UMRON - ".UMR ON" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       FE.EXT OF $FMASK SET IF 11/70
                                ;       NETLDR DATA BLOCK+2 CONTAINS UMR BLOCK ADDRESS (IF ANY)
                                ;
                                ; OUTPUTS:
                                ;       F1.UMR OF $FLAG1 SET
                                ;       ERROR IF 11/70 + NO UMR'S
                                ;-
                                        .IF DF  M$$MGE
000622                          $UMRON::
000622  052767  000000G 000000G         BIS     #F1.UMR,$FLAG1  ; SET THE BIT
                                        .IF DF  R$$11M
000630  032777  000000G 000000G         BIT     #FE.EXT,@FMASK  ; IS THIS REALLY AN 11/70 ?
000636  001417                          BEQ     20$             ; IF EQ, NO
000640  017700  000000G                 MOV     @NTLPT,R0       ; GET DATA BLOCK BIAS

000644                                  SWSTK$  10$             ;* ENTER KERNEL MODE
000650                                  CALL    $MAPX           ;* MAP TO THE DATA BLOCK
000654  016266  000002  000004         MOV     2(R2),4(SP)     ;* PUT UMR BLOCK ADDRESS IN USER R0
                                                                ;* REMEMBER - $MAPX ADDS 1 WORD TO STACK
000662                                  RETURN                  ;* BACK TO USER MODE

000664  005700                  10$:    TST     R0              ; IS THERE A UMR BLOCK ?
000666  001003                          BNE     20$             ; IF NE, YES
000670                                  EMSG$   ZK
                                        .ENDC
000676                          20$:    RETURN
                                        .ENDC
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  EMSG$,EMSG$R,NTLER$

                                ;***
                                ; LOCAL DATA
                                ;***
000000                                  .PSECT  DATA,D

                                ;
                                ; ALLOCATED LIBRARY BLOCK ADDRESS
                                ;
000000                          LBLK::  .BLKW   1

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

                                        .IF DF  M$$MGE
000002                                  NTLER$  ,Y2,6,$TWARN,REP6,$EFNLL,<Not Built For APR*>
                                        .IFF
                                        NTLER$  ,Y2,6,$TWARN,$REP.C,$EFNLL,<Base Address Not Zero>
                                        .IFT
                                        .IF DF  R$$11M
000036                                  NTLER$  ,Y3,6,$TERR,$REP.P,$EFNLL,<Base Address Not a Multiple Of 4K>
                                        .ENDC
                                        .ENDC
000110                                  NTLER$  ,Y4,2,$TERR,$REP.C,$ELINE,<Resource Allocation Failure>
000154                          NTLER$  ,Y6,2,$TERR,$REP.C,$ELINE,<Only 14. Allocations/Libraries Allowed>

                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000234  052     040     124    FMT2::   .ASCIZ  "* Template -- "
000253  052     040     111    FMT6::   .ASCIZ  "* Image File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

                                        .PSECT
```

FILEID**TMPAC8





TMPAC9 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES			
BFADR	000000 R		#5-68	*6-116	6-178	6-194
FMT6	= ******	GX	5-98			
FM.6	= 000000		#5-98			
I$$AS	= ******		6-184			
NLBLK	000002 R		#5-73	*6-165	6-170	
NTLPT	= ******	GX	6-165			
R$$EIS	= ******		6-184			
R$$MPL	= ******		5-75	6-124		
R$$11D	= ******		6-184			
S$$BAS	= ******		5-98	5-98		
$BFALL	= ******	GX	6-179			
$BFBAS	= ******	GX	6-182			
$BFLBN	= ******	GX	6-117	6-118		
$BFXFR	= ******	GX	6-119			
$CLDEQ	= ******	GX	6-122			
$CLIOS	= ******	GX	6-192			
$CUR.N	= ******	GX	5-98			
$EFNLL	= ******	GX	5-98			
$ERRY5	000004 R		#5-98	6-193		
$FNR50	= ******	GX	6-175	6-176	6-177	
$IOSB	= ******	GX	6-192			
$LBADR	= ******	GX	6-174			
$LBIAS	= ******	GX	6-164			
$LBN	= ******	GX	*6-117	*6-118		
$LBXFR	000000 RG		#6-115			
$LLEN	= ******	GX	*6-119			
$MAPX	= ******	GX	6-169			
$READX	= ******	GX	6-120			
$TERR	= ******	GX	5-98			

TMPAC9 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

```
                                40$:    CLC                     ; MAKE SURE THE C-BIT IS CLEAR
                                        .ENDC   ; DF R$$MPL
000436                                  RETURN
                                ;
                                ; ERROR CONDITIONS
                                ;
000440                          101$:   EMSG$R  9C              ; KRB ALLOCATION FAILURE
000446                          111$:   CRASH                   ; LOGIC ERROR - NO SLT ENTRIES ???
```

```
                                BEQ     150$            ;* IF EQ, END OF LIST
                                CMP     #'BS,D.NAM(R1)  ;* IS THIS THE BUS SWITCH DCB ?
                                BNE     30$             ;* IF NE, NO .. KEEP LOOKING
                                MOV     D.UCBL(R1),R2   ;* GET LENGTH OF UCB
                                MOVB    D.UNIT+1(R1),R0 ;* CALCULATE NUMBER OF UNITS-1
                                SUB     D.UNIT(R1),R0   ;*  ...
                                INCB    R0              ;* GET NUMBER OF UNITS
                                MOV     D.UCB(R1),R1    ;* GET ADDRESS OF FIRST UCB
                        40$:    CMP     K.URM-K.PRM(R4),U.CW3(R1) ;* IS THIS THE CONTROLLER'S BUS SWITCH ?
                                BEQ     60$             ;* IF EQ, YES
                                ADD     R2,R1           ;* CALCULATE ADDRESS OF NEXT UCB
                                DECB    R0              ;* EXAMINED ALL UNITS ?
                                BNE     40$             ;* IF NE, NO .. REPEAT
                        60$:    CLR     R0              ;* INITIALIZE CPU UNIBUS RUN MASK
                                MOV     U.SCB(R1),R1    ;* CALCULATE ADDRESS OF KRB TABLE
                                ADD     SKTB,R1         ;*  ...
                        80$:    MOV     (R1)+,R2        ;* EXAMINED EVERY KRB IN TABLE ?
                                BEQ     100$            ;* IF EQ, YES
                                BIS     K.URM(R2),R0    ;* SETUP THE UNIBUS RUN MASK
                                BR      80$             ;* GO AGAIN..
                        100$:   MOV     #-2,R3          ;* INITIALIZE CPU NUMBER*2
                        110$:   ADD     #2,R3           ;* FORM NEXT CPU NUMBER*2
                                ASR     R0              ;* MAY NEED AN ICB FOR HIM?
                                BCC     110$            ;* IF CC, NO
                                CLC                     ;* MAKE SURE THE CARRY IS CLEAR
                                MOV     R0,-(SP)        ;* SAVE REMAINDER OF UNIBUS RUN MASK
                                BIT     $BTMSK(R3),KRB0 ;* ALREADY DID THE ICB'S FOR THIS CPU?
                                BNE     130$            ;* IF NE YES
                                BIS     $BTMSK(R3),KRB0 ;* SHOW AN ICB ALLOCATED FOR THIS CPU
                                MOV     R3,R0           ;* SAVE CPU NUMBER*2
                                SWAB    R0              ;* PUT CPU NUMBER IN ADVERTISED PLACE
                                ASL     R0              ;*
                                ASL     R0              ;*
                                ASL     R0              ;*
                                MOV     R0,CPU          ;*
                                MOV     ICAVL,R0        ;* GET VIRTUAL ADDRESS-2 OF POOL LISTHEAD
                                SUB     #2,R0           ;*  ...
                                MOV     $ICBSZ,R1       ;* GET LENGTH OF AN ICB
                                MOV     K6TAB,K6TMP     ;* GET CPU BIAS TABLE ADDRESS
                                ADD     R3,K6TMP        ;* INDEX INTO TABLE
                                MOV     @K6TMP,@KSAR5   ;* MAP TO CPU PARTITION
                                MOV     #120000,R2      ;* MAP THRU APR5
                                CALL    ALOC2           ;* TRY TO ALLOCATE AN ICB
                                BCS     130$            ;* IF CS, COULD NOT
                                MOVB    $NVECT,(R0)+    ;* STORE THE NUMBER OF I.B'S
                                MOVB    $ICBLN,(R0)+    ;* STORE THE SIZE OF EACH ICB
                                MOV     R0,R1           ;* SAVE VIRTUAL ADDRESS (RELATIVE TO APR5)
                                BIC     #160000,R0      ;* CLEAR APR USEAGE
                                ASR     R0              ;* FORM LINK POINTER
                                ADD     CPU,R0          ;*  ...
                                CLR     (R1)            ;* CLEAR LINK TO NEXT ICB (CLEARS C-BIT)
                                MOV     BIAS,@KSAR5     ;* MAP TO PREVIOUS  ALLOCATION
                                MOV     R0,(R4)         ;* SETUP LINK TO NEXT ICB
                                MOV     R1,R4           ;* SAVE CURRENT ICB ADDRESS
                                MOV     @K6TMP,BIAS     ;* AND BIAS OF CURRENT ICB POOL
                        130$:   MOV     (SP)+,R0        ;* RETORE REMAINDER OF UNIBUS RUN MASK
                                BCS     150$            ;* IF CS, COULD NOT ALLOCATE ICB
```

```
                                ;+
                                ; ERR$N - ERROR MESSAGE SUBROUTINES (PLACED HERE FOR OVERLAY PURPOSES)
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ERROR MESSAGE IS PRINTED
                                ;-
000000                          ERR$1:: EMSG$R  1
000006                          ERR$3:: EMSG$R  3
000014                          ERR$4:: EMSG$R  4
000022                          ERR$5:: EMSG$R  5
000030                          ERR$6:: EMSG$R  6
000036                          ERR$7:: EMSG$R  7
000044                          ERR$9:: EMSG$R  9
000052                          ERR$10:: EMSG$R 10
000060                          ERR$11:: EMSG$R 11
000066                          ERR$12:: EMSG$R 12

        000001                          .END
```

```
                        .TITLE  TMPEX2 - NTL EXPANSION ROUTINES
                        .IDENT  /V05.00/                                ;***-1
;
;COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
;DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
;THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
;ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
;INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
;COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
;OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
;TRANSFERRED.
;
;THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
;AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
;CORPORATION.
;
;DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
;SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
;MODULE DESCRIPTION:
;
;       NTL - STANDARD LINE TABLE EXPANSION ROUTINES
;
;DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
;IDENT HISTORY:
;
;1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
;2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
;3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
;4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
;5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
```

```
                                        ;
                                        ; INTERRUPT ENTRY POINT
                                        ;
001122  012724  013746          INT:    MOV     #13746,(R4)+    ; STORE 'MOV @#KISAR5,-(SP)''
001126  016724  0000C0G                 MOV     KSAR5,(R4)+     ;       ...
001132  012724  012737                  MOV     #12737,(R4)+    ; STORE 'MOV BIAS,@#KISAR5''
001136  016724  000000G                 MOV     $BIAS,(R4)+     ;       ...
001142  016724  000000G                 MOV     KSAR5,(R4)+     ;       ...

001146  012724  004537                  MOV     #4537,(R4)+     ; STORE ''JSR R5,@#ENTRY''
001152  112524                          MOVB    (R5)+,(R4)+     ;       ...
001154  112524                          MOVB    (R5)+,(R4)+     ;       ...
001156                                  RETURN


                                        ;
                                        ; COMPLEMENT OF PRIORITY
                                        ;
001160  012714  000000G         PRI:    MOV     #PR7,(R4)       ; STORE PRIORITY 7
001164  046724  000000G                 BIC     $$PRI,(R4)+     ; CLEAR ACTUAL PRIORITY BITS
001170                                  RETURN
```

TMPEX2 CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 5

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
ASL$	#5-59	10-484								
CALL	7-183	10-451	10-456	11-528	11-533	13-583	13-608	14-647	14-651	15-687
CALLR	15-698									
CRASH	9-295									
DHBDF$	#5-59	5-63								
RESRG	#5-59	9-268								
RETURN	7-185	8-198	8-200	8-210	8-212	8-221	8-230	9-236	9-244	9-250
	9-271	9-274	9-290	9-303	9-311	9-317	9-332	9-334	9-345	9-358
	9-365	9-372	9-383	9-385	9-398	9-400	9-408	9-410	9-418	9-420
	9-427	10-505	11-546	12-561	12-569	13-591	13-623	14-669	15-708	15-718
	15-720	15-727	15-733	16-742	16-747	16-752	16-757	16-762	16-767	16-772
	16-777	16-782	16-787	16-792	17-804	17-814	17-824	17-834	17-844	
SAVRG	#5-59	9-265								
SLTDF$	#5-59	5-64								
SOB	10-499	11-521								
SWSTK$	10-451	11-528	14-647							
TTAB$	#5-73	6-89	6-90	6-91	6-92	6-93	6-94	6-95	6-96	6-97
	6-98	6-99	6-100	6-101	6-102	6-103	6-104	6-105	6-106	6-107
	6-108	6-109	6-110	6-111	6-112	6-113	6-114	6-115	6-116	6-117
	6-118	6-119	6-120	6-121	6-122	6-123	6-124	6-125	6-126	6-127
	6-128	6-129	6-130	6-131	6-132	6-133	6-134	6-135	6-136	6-137
	6-138	6-139	6-140	6-141						

FILEID**TMPEX3

```
                                        .SBTTL  ERROR MESSAGES

                                ;
                                ; ERROR MESSAGES
                                ;

                                        .NLIST  BEX
                                        .ENABL  LC

000020                                  NTLER$  ,23,2,$TERR,$REP.C,$ELINE,<Interrupt Linkage Error>
000060                                  NTLER$  ,24,2B,$TERR,REP2B,$ELINE,<Not In System>
000106                                  NTLER$  ,25,2B,$TERR,REP2B,$ELINE,<Not Available>

000134  052     040     124  FMT2::    .ASCIZ  "* Template -- "
000153  052     040     124  FMT2B::   .ASCIZ  "* Template -- Vector * "

                                        .DSABL  LC
                                        .LIST   BEX
                                        .EVEN
000000                                  .PSECT
```

```
                        .TITLE  TMPAC2 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 2
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

TMPAC2 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 2

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE	REFERENCES
$OFF	= ****** GX	*10-326
$PRV.N	= ****** GX	5-81
$REP.C	= ****** GX	5-82 5-83 5-84 5-85 5-86 5-87 5-88 5-89
$RQ1	= ****** GX	7-203 9-287 10-347 11-373
$RQ2	= ****** GX	10-339
$RQ4	= ****** GX	7-171 9-276
$SECSR	001020 RG	#11-368
$SIZE	= ****** GX	7-176 7-179 *7-180 7-181 *7-202 9-283 9-290 *9-293 10-326 10-343 10-350 *10-352 *11-376
$TERR	= ****** GX	5-81 5-82 5-83 5-84 5-85 5-86 5-87 5-88 5-89
$TMFIX	= ****** GX	6-131
$VALUE	= ****** GX	6-130 *6-133 *7-163 7-165 7-167 7-174 *7-185 *9-262 9-272 9-274 9-279 *9-294 *10-327 10-335 10-337 10-342 *10-353
$VFDDM	= ****** GX	7-159 10-322 11-369
$VFDLC	= ****** GX	9-254
$VFMUX	001070 RG	7-160 10-323 11-370 #12-39[illegible]
$VFSPC	= ****** GX	7-162 9-256 9-266 10-325
$$MTP	= ****** GX	9-262
$$MUX	= ****** GX	7-163 10-327
.EOF	= ****** GX	7-205 9-289 10-349
.FILE	= ****** GX	10-334
.LFILE	= ****** GX	9-270
.MPTAB	= ****** GX	9-260
.MXPTB	= ****** GX	7-173
.MXTAB	= ****** GX	10-328
.SECSR	= ****** GX	11-375

TMPAC2 CREATED BY MACRO ON 29-JUN-85 AT 00:39 PAGE 3

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  ASR$,EMSG$,EMSG$R,NHWDF$,NTLER$

                                ;***
                                ; LIBRARY SYMBOLS
                                ;***
000000                                  NHWDF$

                                ;***
                                ; LOCAL DATA
                                ;***
000000                                  .PSECT  DATA,D

                                ;
                                ; DEFAULT APR NUMBER
                                ;
                                        .IF DF  M$$MGE
        000006                  DFAPR=<<.BASEB/2>&77777>/10000
                                        .ENDC


                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000000                                  NTLER$  ,26,2,$TERR,$REP.C,$ELINE,<Resource Allocation Failure>
                                        .IF DF  M$$MGE
000044                                  NTLER$  ,76,6,$TWARN,REP6,$EFNLL,<Not Built For APR*>
                                        .IFF
                                        NTLER$  ,76,6,$TWARN,$REP.C,$EFNLL,<Base Address Not Zero>
                                        .ENDC
000100                                  NTLER$  ,77,6,$TERR,$CUR.N,$EFNLL,<Read Failure (-***.)>
000136                          NTLER$  ,78,2,$TERR,$REP.C,$ELINE,<Only 14. Allocations/Libraries Allowed>


                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000216  052     040     124     FMT2:   .ASCIZ  "* Template -- "
000235  052     040     111     FMT6:   .ASCIZ  "* Image File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

000000                                  .PSECT
```

```
**FILE**ID**TMPAC5
```

```
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC   5555555555
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC   5555555555
    TT       MMMM  MMMM   PP      PP    AA    AA    CC         55
    TT       MMMM  MMMM   PP      PP    AA    AA    CC         55
    TT       MM  MM  MM   PP      PP    AA    AA    CC         555555
    TT       MM  MM  MM   PP      PP    AA    AA    CC         555555
    TT       MM      MM   PPPPPPPP     AA    AA    CC               55
    TT       MM      MM   PPPPPPPP     AA    AA    CC               55
    TT       MM      MM   PP           AAAAAAAAAA  CC                 55
    TT       MM      MM   PP           AAAAAAAAAA  CC                 55
    TT       MM      MM   PP           AA    AA    CC         55      55       ....
    TT       MM      MM   PP           AA    AA    CC         55      55       ....
    TT       MM      MM   PP           AA    AA      CCCCCCCC   555555        ....
    TT       MM      MM   PP           AA    AA      CCCCCCCC   555555        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

Table of contents


```
     5-   54    MACRO DEFINITIONS
     6-   64    CONSTANT DEFINITIONS
     7-   76    LOCAL DATA
     8-  110    ERROR MESSAGES
     9-  137    $CORE2 - .CORE OPERAND 2
    10-  176    $LIBR1 - .LIBR OPERAND 1
    11-  299    CHECK - CHECK LIBRARY DESCRIPTOR FOR MATCH
    12-  344    $LINKS - .LINKS ACTION ROUTINE
    13-  362    $UMRON - .UMR ON ACTION ROUTINE
    14-  396    $UMROF - .UMR OFF ACTION ROUTINE
    15-  413    $X2CHW - .X2CHW ACTION ROUTINE
    15-  414    $X3CHW - .X3CHW ACTION ROUTINE
    16-  455    $X2CHB - .X2CHB ACTION ROUTINE
    16-  456    $X3CHB - .X3CHB ACTION ROUTINE
    17-  497    $VFLLC - VERIFY LLC
    18-  514    $VXLLC - VERIFY NOT AN LLC
```

```
                                        .SBTTL  $UMROF - .UMR OFF ACTION ROUTINE

                                ;+
                                ; $UMROF - ".UMR OFF" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       F1.UMR IN $FLAG1 IS CLEARED
                                ;-
                                        .IF DF  M$$MGE
000700                          $UMROF::
000700  042767  000000G 000000G         BIC     #F1.UMR,$FLAG1          ;FILE SPACE TO BE ALLOCATED FROM
000706                                  RETURN                          ;  NON-UMR MAPPED MEMORY
                                        .ENDC
```

```
                                ;+
                                ; $LIBR - ".LIBR" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       F2.FIL SET IF VALID FILE NAME HAS BEEN SEEN
                                ;       F2.LBR SET IF LIBRARY WAS ALREADY RESIDENT
                                ;       $LBIAS=LIBRARY DESCRIPTOR BIAS (MAPPED ONLY)
                                ;       $LBADR=LIBRARY DESCRIPTOR ADDRESS OR 0
                                ;       $BFPTR=BINARY BUFFER POINTER
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000000                          $LIBR::
000000  032767  000000G 000000G         BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN
000006  001440                          BEQ     40$             ; IF EQ, NO
000010  032767  000000G 000000G         BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000016  001032                          BNE     30$             ; YES ..
000020  005067  000000'                 CLR     LBLK            ; INDICATE NO LIBRARY BLOCK ALLOCATED
000024  032767  000000G 000000G         BIT     #F2.LBR,$FLAG2  ; WAS LIBRARY ALREADY RESIDENT?
000032  001017                          BNE     20$             ; IF NE, YES
000034  005767  000000G                 TST     $LBADR          ; IS DESCRIPTOR ADDRESS KNOWN?
000040  001003                          BNE     10$             ; IF NE, YES
000042                                  CALL    $ALB            ; ALLOCATE LIBRARY BLOCK
000046  103420                          BCS     40$             ; IF CS, FAILURE
000050                          10$:    CALL    XFER            ; TRANSFER LIBRARY INTO CORE
000054  103415                          BCS     40$             ; IF CS, FAILURE
000056  005767  000000'                 TST     LBLK            ; WAS LIBRARY BLOCK ALLOCATED?
000062  001410                          BEQ     30$             ; IF EQ, NO
000064                                  CALL    LINK            ; LINK THIS BLOCK TO THE OTHERS
000070  000405                          BR      30$
000072                          20$:
                                        .IF DF  M$$MGE
000072                                  CALL    BASE            ; CHECK BASE ADDRESS
                                        .ENDC

000076                                  CALL    MAX14           ; ALLOW ONLY 14. ETC.
000102  103402                          BCS     40$             ; IF CS, FAILURE
000104                          30$:    CALL    USE             ; INCREMENT USE COUNT
000110                          40$:    RETURN
```

```
                .TITLE  TMPAC8 - NTL TEMPLATE PROCESSING ROUTINES
                .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 8
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

TMPAC9 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES				
ASR$	#5-57	6-184			
CALL	6-120	6-122	6-167	6-169	6-193
EMSG$	#5-57	6-193			
NTLER$	#5-57	5-98			
RETURN	6-187	6-195			
SWSTK$	6-167				

FILEID**TMPALT

```
                                        .SBTTL  $TMCHK - CHECK FOR CORRECT DDM INTERRUPT LINKAGE
                                ;+
                                ; $TMCHK - CHECK FOR CORRECT DDM INTERRUPT LINKAGE
                                ;
                                ; INPUTS:
                                ;       $NVECT=NUMBER OF ".INT" OPERATORS PROCESSED
                                ;       $PDVA = PDV ADDRESS OF PROCESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1=DESTROYED
                                ;       F1.ERR SET IF INTERRUPT LINKAGE IS INCORRECT
                                ;-
000450                          $TMCHK::
000450  016700  000000G                MOV     $FLAG1,R0       ; GET TEMPLATE FILE FLAGS
000454  100443                        BMI     30$             ; IF MI, ERRORS - DON'T CHECK
000456  032767  000000G 000000G       BIT     #FL.DDM,$FLAGS  ; ARE WE LOADING A DDM?
000464  001423                        BEQ     10$             ; IF EQ, NO - DON'T CHECK
000466  016701  000000G               MOV     $PDVA,R1        ; GET PDV ADDRESS OF PROCESS
000472  032761  000000G 000000G       BIT     #ZF.PSE,Z.FLG(R1) ; IS IT A PSEUDO DEVICE?
000500  001015                        BNE     10$             ; BR IF YES - DON'T CHECK
000502  016701  000000G               MOV     $NVECT,R1       ; GET NUMBER OF ".INT" OPERATORS
000506  001427                        BEQ     101$            ; IF EQ, NONE SPECIFIED
000510  032700  000000G               BIT     #F1.INT,R0      ; WAS THE ".INT" OPERATOR SEEN?
000514  001007                        BNE     10$             ; IF NE, YES - KNOWN TO BE CORRECT
000516  006301                        ASL     R1              ; MAKE A WORD INDEX
000520  010046                        MOV     R0,-(SP)        ; EXTRACT ".INT" BITS FROM $FLAG1
000522  042716  000000C               BIC     #^C<F1.IN1!F1.IN2!F1.IN3>,(SP)
000526  026126  000002'               CMP     ITAB-2(R1),(SP)+ ; RIGHT COMBINATION OF BITS SET?
000532  001020                        BNE     111$            ; IF NE, NO
000534  032700  000000G       10$:    BIT     #F1.MPL,R0      ; ".MPLHD" OPERATOR SEEN?
000540  001403                        BEQ     20$             ; IF EQ, NO
000542  032700  000000G               BIT     #F1.MPT,R0      ; ".MPTAB" OPERATOR SEEN?
000546  001415                        BEQ     121$            ; IF EQ, NO
000550  032700  000000G       20$:    BIT     #F1.LFL,R0      ; ".LFLHD" OPERATOR SEEN?
000554  001403                        BEQ     30$             ; IF EQ, NO
000556  032700  000000G               BIT     #F1.LFI,R0      ; ".LFILE" OPERATOR SEEN?
000562  001412                        BEQ     131$            ; IF EQ, NO
000564                          30$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
000566                          101$:   EMSG$R  35              ; NO INTERRUPT LINKAGE
000574                          111$:   EMSG$R  36              ; INCOMPLETE INTERRUPT LINKAGE
000602                          121$:   EMSG$R  37              ; ".MPTAB" OPERATOR MISSING
000610                          131$:   EMSG$R  84              ; ".LFILE" OPERATOR MISSING
```

```
1053                                    BNE     110$            ;* IF NE, MORE ICB'S TO ALLOCATE
1054                            150$:   MOV     (SP)+,@KSAR5    ;* RETORE APR MAPPING
1055                                    MOV     #0,R0           ;* LET'S GUARANTEE R0 IS ZERO, BUT SAVE C-BIT !
1056                                    RETURN
1057                                    .ENDC   ; DF R$$MPL
```

```
TMPERR - NTL TEMPLATE ERROR PRO MACRO V05.03b Saturday 29-Jun-85 00:42  Page 6-1
Symbol table

A$$CHK= 000000          ERR$5   000022RG        K$$CSR= 177546          N$$MCP= 000001          X$$DBI= 000000
A$$CPS= 000000          ERR$6   000030RG        K$$LDC= 000000          N$$MLL= 000001          $ELINE= ****** GX
A$$PRI= 000000          ERR$7   000036RG        K$$TPS= 000074          N$$MOV= 000010          $ERR1   000144RG    002
A$$TRP= 000000          ERR$9   000044RG        LD$LP = 000000          N$$NCT= 000001          $ERR10  000026RG    002
C$$CKP= 000000          E$$XPR= 000000          L$$ASG= 000000          N$$PEM= 000001          $ERR11  000064RG    002
C$$ORE= 000400          FMT2    000420R  002    L$$DRV= 000000          P$$P45= 000000          $ERR12  000116RG    002
C$$RSH= 177564          FMT3    000437R  002    L$$P11= 000001          P$$WRD= 000000          $ERR3   000200RG    002
D$$BUG= 177514          FM.2  = 000000          L$$11R= 000000          Q$$OPT= 000010          $ERR4   000234RG    002
D$$ISK= 000000          FM.3  = 000000          M$$CRB= 000124          R$$DER= 000000          $ERR5   000276RG    002
D$$L11= 000001          F$$LVL= 000001          M$$CRX= 000000          R$$K11= 000001          $ERR6   000340RG    002
D$$YNC= 000000          G$$TPP= 000000          M$$FCS= 000000          R$$SND= 000000          $ERR7   000370RG    002
D$$YNM= 000000          G$$TSS= 000000          M$$MGE= 000000          R$$11M= 000000          $ERR9   000000RG    002
ERR$1   000000RG        G$$TTK= 000000          M$$NET= 000000          S$$WRG= 000000          $PRV.N= ****** GX
ERR$10  000052RG        G$$WRD= 000000          M$$OVR= 000000          S$$YSZ= 007600          $REP.C= ****** GX
ERR$11  000060RG        I$$RAR= 000000          N$$ACC= 000001          T$$KMG= 000000          $REP.P= ****** GX
ERR$12  000066RG        I$$RDN= 000000          N$$BUF= 000001          T$$MIN= 000000          $TERR = ****** GX
ERR$3   000006RG        K$$CNT= 177546          N$$LDV= 000001          V$$CTR= 001000          $TWARN= ****** GX
ERR$4   000014RG

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000074    001   (RW,I,LCL,REL,CON)
DATA    000464    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10107 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:10.11
SY:TMPERR.V2,[132,134]TMPERR/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPERR
```

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  ASL$,DHBDF$,SLTDF$,SAVRG,RESRG
                                ;***
                                ; DEFINE OFFSETS
                                ;***
000000                                  DHBDF$                  ; DEFINE DECNET HOME BLOCK OFFSETS
000000                                  SLTDF$                  ; DEFINE SLT OFFSETS

                                ;***
                                ; LOCAL MACROS
                                ;***

                                ;
                                ; DEFINE TRANSFER TABLE ENTRY
                                ;
                                        .MACRO  TTAB$   AA,BB
                                        .WORD   AA
                                        .ENDM   TTAB$
```

```
                                        ;
                                        ; SECONDARY FILE
                                        ;
001172  012767  100000  000154' MXTAB:  MOV     #100000,FLAGS   ; INDICATE MUX EXPANSION
001200  112546                  FILE:   MOVB    (R5)+,-(SP)     ; GET REPEAT COUNT
001202  010546                          MOV     R5,-(SP)        ; SAVE CURRENT BINARY FILE POINTER
001204  011605                  10$:    MOV     (SP),R5         ; RESTORE BINARY FILE POINTER
001206  005003                  20$:    CLR     R3              ; GET FUNCTION CODE
001210  152503                          BISB    (R5)+,R3        ;       ...
001212  122703  000000G                 CMPB    #.EOF,R3        ; IS IT END OF FILE?
001216  001404                          BEQ     30$             ; IF EQ, YES
001220  106303                          ASLB    R3              ; CONVERT TO A WORD INDEX
001222                                  CALL    @TABLE-2(R3)    ; EXPAND A FIELD
001226  000767                          BR      20$             ; KEEP LOOKING FOR EOF
001230  005267  000152'        30$:    INC     INDEX           ; TALLY ONE PASS THRU SECONDARY FILE
001234  126667  000002  000152'        CMPB    2(SP),INDEX     ; ARE WE DONE?
001242  001360                          BNE     10$             ; IF NE, NO
001244  022626                          CMP     (SP)+,(SP)+     ; PURGE STACK
001246  005067  000152'                 CLR     INDEX           ; CLEAR SECONDARY INDEX
001252  005067  000154'                 CLR     FLAGS           ; AND SPECIAL FLAGS
001256                                  RETURN
                                        ;
                                        ;
                                        ; LINKED SECONDARY FILE
                                        ;
001260  012767  000200  000154' MPTAB:  MOV     #200,FLAGS      ; INDICATE MTP EXPANSION
001266  112546                  LFILE:  MOVB    (R5)+,-(SP)     ; SAVE REPEAT COUNT
001270  112546                          MOVB    (R5)+,-(SP)     ; AND LISTHEAD DISPLACEMENT
001272  112566  000001                  MOVB    (R5)+,1(SP)     ;       ...
001276  010546                          MOV     R5,-(SP)        ; SAVE BINARY POINTER
001300  011605                  10$:    MOV     (SP),R5         ; RESET BINARY POINTER
001302  010446                          MOV     R4,-(SP)        ; SAVE SECONDARY TABLE ADDRESS
001304  005003                  20$:    CLR     R3              ; GET FUNCTION CODE
001306  152503                          BISB    (R5)+,R3        ;       ...
001310  022703  000000G                 CMP     #.EOF,R3        ; IS IT END OF FILE?
001314  001404                          BEQ     30$             ; IF EQ, YES
001316  106303                          ASLB    R3              ; CONVERT TO WORD INDEX
001320                                  CALL    @TABLE-2(R3)    ; EXPAND A FIELD
001324  000767                          BR      20$             ; KEEP LOOKING FOR EOF
001326  016700  000000G         30$:    MOV     $ADDR,R0        ; GET MULTIPOINT LISTHEAD ADDRESS
001332  066600  000004                  ADD     4(SP),R0        ;       ...
001336  012603                          MOV     (SP)+,R3        ; RESTORE SECONDARY TABLE ADDRESS
001340  016002  000002                  MOV     2(R0),R2        ; LINK SECONDARY TABLE TO LISTHEAD
001344  010360  000002                  MOV     R3,2(R0)        ;       ...
001350  005013                          CLR     (R3)            ;       ...
001352  010312                          MOV     R3,(R2)         ;       ...
001354  005267  000152'                 INC     INDEX           ; TALLY ONE PASS THRU SECONDARY FILE
001360  126667  000004  000152'        CMPB    4(SP),INDEX     ; ARE WE DONE?
001366  001344                          BNE     10$             ; IF NE, NO
001370  062706  000006                  ADD     #6,SP           ; PURGE STACK
001374  005067  000152'                 CLR     INDEX           ; CLEAR SECONDARY INDEX
001400  005067  000154'                 CLR     FLAGS           ; AND SPECIAL FLAGS
001404                                  RETURN
```

```
**FILE**ID**TMPEX3
```

```
TTTTTTTTTT   MM      MM   PPPPPPPP     EEEEEEEEEE   XX      XX     333333
TTTTTTTTTT   MM      MM   PPPPPPPPP    EEEEEEEEEE   XX      XX     333333
    TT       MMMM  MMMM   PP      PP   EE           XX      XX   33      33
    TT       MMMM  MMMM   PP      PP   EE           XX      XX   33      33
    TT       MM  MM  MM   PP      PP   EE             XX  XX               33
    TT       MM  MM  MM   PP      PP   EE             XX  XX               33
    TT       MM      MM   PPPPPPPP     EEEEEEEE         XX             33
    TT       MM      MM   PPPPPPPP     EEEEEEEE         XX             33
    TT       MM      MM   PP           EE             XX  XX               33
    TT       MM      MM   PP           EE             XX  XX               33
    TT       MM      MM   PP           EE           XX      XX   33      33      ....
    TT       MM      MM   PP           EE           XX      XX   33      33      ....
    TT       MM      MM   PP           EEEEEEEEEE   XX      XX     333333        ....
    TT       MM      MM   PP           EEEEEEEEEE   XX      XX     333333        ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .SBTTL  ACTION ROUTINES
                                ;+
                                ; $INTX - ".INT" ACTION ROUTINE, PART 2
                                ;
                                ; INPUTS:
                                ;       $VALUE=ENTRY POINT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;-
000000                          $INTX::
000000                                  CALL    $RQ3            ; REQUEST BINARY FILE
000004  103411                         BCS     10$             ; IF CS, NO ACCESS
000006  112722  000000G                MOVB    #.INT,(R2)+     ; STORE FUNCTION CODE
000012  116722  000000G                MOVB    $VALUE,(R2)+    ; STORE ENTRY POINT ADDRESS
000016  116712  000001G                MOVB    $VALUE+1,(R2)   ;       ...
000022  062767  000016  000000G        ADD     #14.,$SIZE      ; ADJUST DATA BASE SIZE
000030                          10$:    RETURN
```



```
                                        .SBTTL  ACTION ROUTINES
                                ;+
                                ; $INTX - ".INT" ACTION ROUTINE, PART 2
                                ;
                                ; INPUTS:
                                ;       $VALUE=ENTRY POINT ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;-
000000                          $INTX::
000000                                  CALL    $RQ3            ; REQUEST BINARY FILE
000004  103411                         BCS     10$             ; IF CS, NO ACCESS
000006  112722  000000G                MOVB    #.INT,(R2)+     ; STORE FUNCTION CODE
000012  116722  000000G                MOVB    $VALUE,(R2)+    ; STORE ENTRY POINT ADDRESS
000016  116712  000001G                MOVB    $VALUE+1,(R2)   ;       ...
000022  062767  000016  000000G        ADD     #14.,$SIZE      ; ADJUST DATA BASE SIZE
000030                          10$:    RETURN
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  EMSG$,EMSG$R,NTLER$

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                        .NLIST  BEX

                                ;
                                ; TEMPLATE EXTENSION
                                ;
000000  104  101  124  TMPEXT: .ASCIZ  "DAT"

                                        .EVEN


                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000004                                  NTLER$  ,17,3,$TERR,$PRV.N,$EFNLL,<Open Failure (-***.)>
000042                                  NTLER$  ,41,2,$TERR,$REP.C,$ELINE,<".MPLHD" Operator Missing>
000104                                  NTLER$  ,42,2,$TERR,$REP.C,$ELINE,<".LFLHD" Operator Missing>
000146                                  NTLER$  ,43,2,$TERR,$REP.C,$ELINE,<Secondary Table Empty>
000204                                  NTLER$  ,22,2,$TERR,$REP.C,$ELINE,<File Count Too Large>
000242                                  NTLER$  ,81,2,$TERR,$REP.C,$ELINE,<MUX Operator>
000270                                  NTLER$  ,85,2,$TERR,$REP.C,$ELINE,<KMC Operator>
000316                                  NTLER$  ,KX,2,$TERR,$REP.C,$ELINE,<Resource Allocation Failure>
000362                          NTLER$  ,KY,2,$TERR,$REP.C,$ELINE,<Only 14. Allocations/Libraries Allowed>

                                        .DSABL  LC


                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000442  052  040  124  FMT2:   .ASCIZ  "* Template -- "
000461  052  040  124  FMT3:   .ASCIZ  "* Template File -- "
                                .EVEN
                                .DSABL  LC
                                .LIST   BEX

000000                          .PSECT
```

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
CALL	6-126	6-127	7-159	7-160	7-162	7-169	7-171	7-177	7-178	7-195
	7-203	9-254	9-256	9-266	9-276	9-284	9-285	9-287	10-322	10-323
	10-325	10-329	10-333	10-339	10-344	10-345	10-347	11-369	11-370	11-373
	12-396									
CALLR	6-131									
EMSG$	#5-59	12-396								
EMSG$R	#5-59	6-139	7-210	7-211	8-237	9-292	9-301	9-302	9-303	11-382
NTLER$	#5-59	5-81	5-82	5-83	5-84	5-85	5-86	5-87	5-88	5-89
RETURN	6-134	7-206	8-233	9-296	10-331	10-355	11-377	12-397		

FILEID**TMPAC3

```
                                ;+
                                ; $BINF - ".BIN" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       BINARY FILE IS OPEN (OPENED BY $ABFI)
                                ;       $BFALL=ALLOCATION SIZE
                                ;       $BFLBN=LOGICAL BLOCK NUMBER
                                ;       $BFPTR=BINARY BUFFER POINTER
                                ;       $BFXFR=TRANSFER SIZE IN BYTES
                                ;       F1.UMR IN $FLAG1 INDICATES WHETHER MEMORY SHOULD BE ALLOCATED
                                ;         FROM UMR OR NON-UMR MEMORY
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000000                          $BINF::
000000  032767  000000G 000000G          BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN?
000006  001505                          BEQ     30$             ; IF EQ, NO
000010  005000                          CLR     R0              ; NO ALLOCATION ADDRESS YET
000012  032767  000000G 000000G          BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000020  001030                          BNE     22$             ; YES .. ($BFALL IS ZERO)
                                
                                        .IF DF  M$$MGE
                                        .IF DF  R$$11M
000022  026767  000000G 000000G          CMP     $BFBAS,$VALUE   ; BUILT FOR CORRECT APR?
                                        .IFF
                                        CMPB    $BFBAS,$CAPR    ; BUILT FOR CORRECT APR?
                                        .ENDC
                                        .IFF
                                        TST     $BFBAS          ; BUILT FOR BASE ADDRESS OF 0?
                                        .ENDC
                                
000030  001403                          BEQ     10$             ; IF EQ, YES
000032                                  EMSG$   76              ; ELSE PRINT WARNING MESSAGE
000040                          10$:    CALL    MAX14           ; ALLOW ONLY 14. ALLOCATIONS
000044  103464                          BCS     25$             ; IF CS, NO MORE ALLOWED
000046  016701  000000G                 MOV     $BFALL,R1       ; GET ALLOCATION SIZE NEEDED
                                
                                        .IF DF  M$$MGE
000052  032767  000000G 000000G          BIT     #F1.UMR,$FLAG1  ; ALLOCATE UMR OR NON-UMR MEMORY
000060  001003                          BNE     15$             ; BR IF ALLOCATE UMR MEMORY
000062                                  CALL    $ALN18          ; ALLOCATE FROM NON-UMR SPACE
000066  000402                          BR      20$             ; AND PROCESS
                                        .ENDC
                                
000070                          15$:    CALL    $ALL18          ; TRY TO ALLOCATE FROM UMR SPACE
000074  103453                  20$:    BCS     101$            ; BR IF ALLOCATION FAILURE
000076  105267  000000G                 INCB    $NALL           ; TALLY SUCCESSFUL ALLOCATION
000102  016702  000000G          22$:    MOV     $BFPTR,R2       ; RESTORE BINARY BUFFER POINTER
000106  112722  000000G                 MOVB    #.CORE,(R2)+    ; STORE FUNCTION CODE
000112  110022                          MOVB    R0,(R2)+        ; FIRST WORD OF ADDRESS
000114  000300                          SWAB    R0              ;       ...
000116  110022                          MOVB    R0,(R2)+        ;       ...
                                
                                        .IF DF  M$$MGE
000120  116722  000000G                 MOVB    $VALUE,(R2)+    ; SECOND WORD OF ADDRESS
000124  116722  000001G                 MOVB    $VALUE+1,(R2)+  ;       ...
```

```
                        .TITLE  TMPAC5 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 5
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
                        .TITLE  TMPAC6 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/

;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 6
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
```

```
                                        .SBTTL  $X2CHW - .X2CHW ACTION ROUTINE
                                        .SBTTL  $X3CHW - .X3CHW ACTION ROUTINE

                                ;+
                                ;
                                ; $X2CHW - .X2CHW ACTION ROUTINE
                                ; $X3CHW - .X3CHW ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $VALUE - VALUE OF FIRST OPERAND (INDICATES WHICH PARAMETER
                                ;                FROM X2P$DF OR X3P$DF TO USE)
                                ;
                                ; OUTPUTS:
                                ;       R1 DESTROYED
                                ;
                                ;-
                                        .ENABL  LSB
000710  012701  000000G         $X2CHW::MOV     #.X2CH1,R1              ; GET FUNCTION CODE
000714  026727  000000G 000002          CMP     $VALUE,#X2BLK           ; LEVEL 2 BLOCK SIZE?
000722  001410                          BEQ     5$                      ; YES, SPECIAL CASE (NOT IN RANGE)
000724  026727  000000G 000004          CMP     $VALUE,#X2WLO           ; IS VALUE WITHIN RANGE?
000732  103421                          BLO     10$                     ; BR IF NO
000734  026727  000000G 000005          CMP     $VALUE,#X2WHI           ; WITHIN RANGE?
000742  101015                          BHI     10$                     ; BR IF NO
000744                          5$:     CALL    $VXLLC                  ; MAKE SURE IT'S NOT AN LLC
000750  000417                          BR      30$

000752  012701  000000G         $X3CHW::MOV     #.X3C01,R1              ; GET FUNCTION CODE
000756  026727  000000G 000001          CMP     $VALUE,#X3WLO           ; IS OPERAND WITHIN RANGE?
000764  103404                          BLO     10$                     ; BR IF NO
000766  026727  000000G 000002          CMP     $VALUE,#X3WHI           ; WITHIN RANGE?
000774  101403                          BLOS    20$                     ; BR IF YES
000776                          10$:    EMSG$   91                      ; INVALID OPERAND
001004                          20$:    CALL    $VFLLC                  ; MAKE SURE IT IS AN LLC
001010  066701  000000G                ADD     $VALUE,R1               ; POINT TO CORRECT FUNCTION CODE
001014  005301                          DEC     R1                      ; OPERAND STARTS AT 1
001016                                  CALL    $RQ1                    ; REQUEST BINARY BUFFER
001022  110122                          MOVB    R1,(R2)+                ; STORE FUNCTION CODE
001024  062767  000002  000000G         ADD     #2,$SIZE                ; ADJUST DATA BASE SIZE
001032                                  RETURN
                                        .DSABL  LSB
```

```
                                ;+
                                ; XFER - TRANSFER LIBRARY FILE INTO CORE
                                ;
                                ; INPUTS:
                                ;       $BFALL=ALLOCATION SIZE NEEDED
                                ;       $BFLBN=LOGICAL BLOCK NUMBER
                                ;       $BFXFR=TRANSFER SIZE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000112                          XFER:
                                        .IF DF  M$$MGE
                                        .IF DF  R$$11M
000112  032767  017777  000000G         BIT     #17777,$BFBAS   ; BUILT TO SOME 4K BOUNDARY?
000120  001032                          BNE     101$            ; IF NE, NO
000122  026767  000000G 000000G         CMP     $VALUE,$BFBAS   ; BUILT FOR CORRECT APR?
                                        .IFF
                                        CMPB    $BFBAS,$CAPR    ; DITTO?
                                        .ENDC
                                        .IFF
                                        TST     $BFBAS          ; BUILT FOR BASE ADDRESS OF 0?
                                        .ENDC

000130  001403                         BEQ     10$             ; IF EQ, YES
000132                                  EMSG$   Y2              ; PRINT WARNING MESSAGE
000140                          10$:    CALL    MAX14           ; ALLOW ONLY 14. ALLOCS/LIBRS
000144  103417                         BCS     25$             ; IF CS, TOO MANY ALREADY
000146  016701  000000G                MOV     $BFALL,R1       ; GET ALLOCATION SIZE NEEDED

                                        .IF DF  M$$MGE
000152  032767  000000G 000000G         BIT     #F1.UMR,$FLAG1  ; MUST FILE BE IN UMR SPACE?
000160  001003                          BNE     15$             ; BR IF YES
000162                                  CALL    $ALN18          ; ALLOCATE FROM NON-UMR SPACE
000166  000402                         BR      20$             ; AND PROCESS
                                        .ENDC

000170                          15$:    CALL    $ALL18          ; ALLOCATE FROM NT POOL (UMR SPACE)
000174  103411                 20$:    BCS     111$            ; BR IF ALLOCATION FAILURE
000176                                  CALL    $LBXFR          ; TRANSFER LIBRARY INTO CORE
000202  103413                         BCS     121$            ; IF CS, FAILURE
000204                          25$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
                                        .IF DF  R$$11M & M$$MGE
000206                          101$:   CALL    $LBERR          ; CLEANUP AFTER AN ERROR
000212                                  EMSG$R  Y3              ; LIBRARY BASE ADDRESS NOT MULTIPLE OF 4K
                                        .ENDC
000220                          111$:   CALL    $LBERR          ; CLEANUP AFTER AN ERROR
000224                                  EMSG$R  Y4              ; LIBRARY ALLOCATION FAILURE
000232  016701  000000G         121$:   MOV     $BFALL,R1       ; GET LIBRARY SIZE
000236                                  CALL    $DEA18          ; DE-ALLOCATE LIBRARY
000242                                  CALL    $LBERR          ; CLEANUP AFTER AN ERROR
000246  000261                         SEC                     ; RETURN C-BIT SET
000250                                  RETURN
```

```
                        ;***
                        ; LIBRARY MACROS
                        ;***
                                .MCALL  EMSG$R,NTLER$

                        ;***
                        ; LOCAL DATA
                        ;***
000000                          .PSECT  DATA,D

                        ;
                        ; ERROR MESSAGE (FORMAT STRING ADDRESS DEFINED GLOBALLY IN ''TMPAC7'')
                        ;
                                .ENABL  LC
000000                          NTLER$  ,Y1,2,$TERR,$REP.C,$ELINE,<Library Block Allocation Failure>
                                .DSABL  LC

                        ;
                        ; 32. WORD BLOCK ALLOCATION SIZE
                        ;
000001                          .IIF DF M$$MGE, BLKSIZ=1
                                .IIF NDF M$$MGE, BLKSIZ=100

000000                          .PSECT
```

```
**FILE**ID**TMPALT
```

```
TTTTTTTTTT   MM       MM   PPPPPPPP       AAAAAA     LL           TTTTTTTTTT
TTTTTTTTTT   MM       MM   PPPPPPPP      AAAAAA      LL           TTTTTTTTTT
    TT       MMMM   MMMM   PP      PP   AA      AA   LL               TT
    TT       MMMM   MMMM   PP      PP   AA      AA   LL               TT
    TT       MM  MM  MM   PP      PP   AA      AA   LL               TT
    TT       MM  MM  MM   PP      PP   AA      AA   LL               TT
    TT       MM      MM   PPPPPPPP     AA      AA   LL               TT
    TT       MM      MM   PPPPPPPP     AA      AA   LL               TT
    TT       MM      MM   PP           AAAAAAAAAA   LL               TT
    TT       MM      MM   PP           AAAAAAAAAA   LL               TT
    TT       MM      MM   PP           AA      AA   LL               TT          ....
    TT       MM      MM   PP           AA      AA   LL               TT          ....
    TT       MM      MM   PP           AA      AA   LLLLLLLLLL       TT          ....
    TT       MM      MM   PP           AA      AA   LLLLLLLLLL       TT          ....
```

```
LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                                .SBTTL  UBA - GET UNLOAD BLOCK ADDRESS
                                        ;+
                                        ; UBA - CALCULATE UNLOAD BLOCK ADDRESS
                                        ;
                                        ; INPUTS:
                                        ;       $ADDR   = LINE TABLE ADDRESS (OR BIAS)
                                        ;       $SIZE   = LINE TABLE SIZE (IN BYTES OR BLOCKS)
                                        ;       $NALL   = NUMBER OF ALLOCATIONS (BYTE FIELD)
                                        ;       $NLIB   = NUMBER OF LIBRARIES (BYTE FIELD)
                                        ;       $NTLPT  = POINTER TO NTL DATA BLOCK
                                        ;       $NALL,$NLIB = A WORD FIELD
                                        ;       DATA BLOCK+.CXUNL = POINTER TO FIRST UNLOAD BLOCK
                                        ;
                                        ; OUTPUTS:
                                        ;       $UBIAS  = BIAS OF UNLOAD BLOCK (MAPPED SYSTEM ONLY)
                                        ;       $UDA    = UNLOAD DATA ADDRESS FOR FIRST ALLOCATION ENTRY
                                        ;       $LDA    = UNLOAD DATA ADDRESS FOR FIRST LIBRARY DESCRIPTOR ENTRY
                                        ;       $SIZE   = DESTROYED IF DATA BASE WAS MAPPED
                                        ;-
000616                                  UBA:    CALL    GETIND          ; GET PDV INDEX OF CURRENT PROCESS
000622  005067  000000G                        CLR     $UDA            ; SET NO UNLOAD DATA ADDRESS YET
000626  116700  000000G                        MOVB    $NALL,R0        ; GET NUMBER OF ALLOCATIONS
000632  116701  000000G                        MOVB    $NLIB,R1        ; AND NUMBER OF LIBRARIES
000636  060001                                ADD     R0,R1           ; ADD THEM TOGETHER
000640  001403                                BEQ     10$             ; IF ZERO, NO ALLOCS/LIBRS
000642  006301                                ASL     R1              ; MULTIPLY BY 4 (2 WORDS PER UNLOAD ENTRY)
000644  006301                                ASL     R1              ;
000646  005721                                TST     (R1)+           ; INCLUDE 1 WORD FOR ALLOC/LIBR COUNT
000650  022121                        10$:    CMP     (R1)+,(R1)+     ; INCLUDE 2 WORDS FOR LINE TABLE ADDRESS/SIZE
000652  005721                                TST     (R1)+           ; INCLUDE 1 WORD FOR PDV INDEX AND RESERVED BYTE
000654  010167  000000G                        MOV     R1,$LDA         ; SAVE UNLOAD SIZE NEEDED
                                        ;
                                                .IF DF  M$$MGE
                                        ;
000660                                          SWSTK$  70$             ;* ENTER KERNEL MODE
000664  017700  000000G                        MOV     @NTLPT,R0       ;* GET DATA BLOCK ADDRESS
000670  010067  000604'                        MOV     R0,NLBLK        ;* SAVE IT FOR LATER
000674                                          CALL    $MAPX           ;* MAP TO THE DATA BLOCK
000700  016211  000000G                        MOV     .CXUNL(R2),(R1) ;* MAP TO FIRST UNLOAD BLOCK
                                        ;
                                                .IFF
                                        ;
                                                MOV     $NTLPT,R0       ; GET DATA BLOCK ADDRESS
                                                MOV     .CXUNL(R0),R2   ; GET FIRST UNLOAD BLOCK ADDRESS
                                        ;
                                                .IFTF
                                        ;
000704  001437                                  BEQ     60$             ;* IF EQ, NO UNLOAD BLOCKS YET
000706  010200                        20$:      MOV     R2,R0           ;* COPY VIRTUAL ADDRESS
000710  062700  000100                          ADD     #100,R0         ;* SET 32. WORD BLOCK VIRTUAL ADDRESS LIMIT
000714  010203                                  MOV     R2,R3           ;* COPY VIRTUAL ADDRESS AGAIN
000716  005723                                  TST     (R3)+           ;* SKIP NEXT UNLOAD BLOCK POINTER
000720  020300                        30$:      CMP     R3,R0           ;* IS IT END OF THIS 32. WORD BLOCK?
000722  103026                                  BHIS    50$             ;* IF HIS, YES
000724  005723                                  TST     (R3)+           ;* SKIP THE PDV INDEX ENTRY (IF EQ, COULD BE AUX)
000726  005723                                  TST     (R3)+           ;* SKIP LINE TABLE ADDRESS ENTRY
000730  001412                                  BEQ     40$             ;* IF ZERO, THE REST OF THIS BLOCK IS AVAILABLE
```

```
1059                            .SBTTL  ALOC2 - ALLOCATE FROM OUTSIDE TASK SPACE
1060                    ;+
1061                    ; *** - ALOC2 - ALLOCATE POOL SPACE FROM AREA OUTSIDE TASK'S MAPPING SPACE
1062                    ;
1063                    ; THIS ROUTINE WAS ADAPTED FROM THE RSX-11M EXECUTIVE ROUTINE $ALOC1.
1064                    ; ADDITIONAL CODE WAS ADDED FOR REMAPPING OF THE ADDRESSES IN THE CHAIN.
1065                    ; FOR ADDITIONAL EXPLANATION OF THIS ROUTINE, CONSULT THE EXECUTIVE
1066                    ; MODULE "CORAL".
1067                    ; THIS ROUTINE ALLOCATES POOL SPACE FROM MEMORY THAT IS OUTSIDE THE
1068                    ; MAPPING SPACE OF THE TASK. THE APR MUST HAVE ALREADY BEEN SET UP BEFORE
1069                    ; CALLING THIS ROUTINE, WITH THE DISPLACEMENT FOR THE APR PASSED IN R2.
1070                    ; THIS ROUTINE ASSUMES THAT THE POINTERS IN THE LINKED LIST ARE BIASED BY
1071                    ; APR6. THIS ROUTINE WILL ENDEAVOR TO PRESERVE THE APR6 MAPPING.
1072                    ;
1073                    ; INPUTS:
1074                    ;       R0 = ADDRESS OF ALLOCATION LISTHEAD-2 (UNBIASED)
1075                    ;       R1 = LENGTH OF CORE NEEDED IN BYTES
1076                    ;       R2 = DISPLACEMENT ADDRESS (100000 OR 120000)
1077                    ;
1078                    ; OUTPUTS:
1079                    ;       C-BIT   SET IF NO MEMORY ALLOCATED
1080                    ;               CLEAR IF MEMORY WAS ALLOCATED
1081                    ;
1082                    ;       IF SUCCESS:
1083                    ;               R0 = ADDRESS OF THE ALLOCATED BLOCK.
1084                    ;               R1 = LENGTH OF THE ALLOCATED BLOCK.
1085                    ;
1086                    ;-
1087
1088                            .IF DF  R$$MPL
1089                    ALOC2:
1090                            BIS     R2,R0           ;LETS MAP TO CPU PARTITION
1091                            ADD     (R0),R1         ;ROUND TO NEXT BOUNDARY
1092                            BIC     (R0)+,R1        ;CLEAR EXCESS
1093                            SEC                     ;ASSUME ZERO LENGTH BLOCK
1094                            BEQ     30$             ;IF EQ ZERO LENGTH REQUEST
1095                            MOV     R2,-(SP)        ;SAVE RELOCATION ADDRESS
1096                    10$:    MOV     R0,R2           ;SAVE ADDRESS OF CURRENT BLOCK
1097                            MOV     (R2),R0         ;GET ADDRESS OF NEXT BLOCK
1098                            BEQ     25$             ;IF EQ END OF CHAIN
1099                            BIC     #160000,R0      ;CLEAR CURRENT MAPPING REG.
1100                            BIS     (SP),R0         ;USE DUR MAPPING REG.
1101                            CMP     2(R0),R1        ;BLOCK BIG ENOUGH?
1102                            BLO     10$             ;IF LO NO
1103                            BEQ     20$             ;IF EQ BLOCK IS EXACT SIZE
1104                            TST     (R0)+           ;POINT TO SIZE OF FREE BLOCK
1105                            SUB     R1,(R0)         ;CALCULATE SIZE REMAINING
1106                            ADD     R0,R1           ;POINT TO NEW FREE BLOCK
1107                            MOV     (R0),(R1)       ;COPY SIZE
1108                            MOV     -(R0),-(R1)     ;COPY LINK
1109                            BIC     #160000,R1      ;GET RID OF MAPPING REGISTER
1110                            MOV     R1,(R0)         ;STORE LINK TO NEW FREE BLOCK
1111                    20$:    MOV     (R0),(R2)       ;UPDATE LINK TO NEW FREE BLOCK
1112                    25$:    MOV     (SP)+,R2        ;RETORE RELOCATION ADDRESS
1113                    30$:    RETURN
1114                            .ENDC   ; DF R$$MPL
```

TMPERR CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
ERR$1	000000	RG	#6-110								
ERR$10	000052	RG	#6-117								
ERR$11	000060	RG	#6-118								
ERR$12	000066	RG	#6-119								
ERR$3	000006	RG	#6-111								
ERR$4	000014	RG	#6-112								
ERR$5	000022	RG	#6-113								
ERR$6	000030	RG	#6-114								
ERR$7	000036	RG	#6-115								
ERR$9	000044	RG	#6-116								
FMT2	000420	R	5-71	5-79	5-80	5-81	5-82	5-83	5-84	#5-93	
FMT3	000437	R	5-72	5-73	5-74	#5-94					
FM.2	= 000000		#5-71	#5-79	#5-80	#5-81	#5-82	#5-83	#5-84		
FM.3	= 000000		#5-72	#5-73	#5-74						
S$$BAS	= ******		5-71	5-71	5-72	5-72	5-73	5-73	5-74	5-74	5-79
			5-79	5-80	5-80	5-81	5-81	5-82	5-82	5-83	5-83
			5-84	5-84							
$ELINE	= ******	GX	5-71	5-74	5-79	5-80	5-81	5-82	5-83	5-84	
$ERR1	000144	RG	#5-79	6-110							
$ERR10	000026	RG	#5-72	6-117							
$ERR11	000064	RG	#5-73	6-118							
$ERR12	000116	RG	#5-74	6-119							
$ERR3	000200	RG	#5-80	6-111							
$ERR4	000234	RG	#5-81	6-112							
$ERR5	000276	RG	#5-82	6-113							
$ERR6	000340	RG	#5-83	6-114							
$ERR7	000370	RG	#5-84	6-115							
$ERR9	000000	RG	#5-71	6-116							
$PRV.N	= ******	GX	5-72								
$REP.C	= ******	GX	5-71	5-79	5-80	5-81	5-82	5-83	5-84		
$REP.P	= ******	GX	5-73	5-74							
$TERR	= ******	GX	5-71	5-72	5-74	5-79	5-80	5-81	5-82	5-83	5-84
$TWARN	= ******	GX	5-73								

TMPERR CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 2

LOCAL DATA

```
                                        .SBTTL  LOCAL DATA

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                ;
                                ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                                ;
                                TABLE:  TTAB$   ADDRB,NULL      ; EXPANSION ROUTINE IS FIRST ENTRY
000002                                  TTAB$   ADDRW,NULL      ; CLEANUP ROUTINE IS SECOND ENTRY
000004                                  TTAB$   BLKB,POP1
000006                                  TTAB$   BLKW,POP1
000010                                  TTAB$   BYTE,POP1
000012                                  TTAB$   CORE,XCORE
000014                                  TTAB$   CSR,POP1
000016                                  TTAB$   CTIM,NULL
000020                                  TTAB$   DVCHA,NULL
000022                                  TTAB$   EOF,NULL
000024                                  TTAB$   FILE,POP1
000026                                  TTAB$   INT,POP2
000030                                  TTAB$   LIBR,XLIBR
000032                                  TTAB$   LINKS,NULL
000034                                  TTAB$   LFILE,POP3
000036                                  TTAB$   LSTHD,NULL
000040                                  TTAB$   LTAB,NULL
000042                                  TTAB$   MPTAB,POP3
000044                                  TTAB$   MXTAB,POP1
000046                                  TTAB$   NOD,NULL
000050                                  TTAB$   PRI,NULL
000052                                  TTAB$   PRIOV,POP1
000054                                  TTAB$   SCOM,XSCOM
000056                                  TTAB$   SECSR,NULL
000060                                  TTAB$   SLNB,NULL
000062                                  TTAB$   SLNW,NULL
000064                                  TTAB$   STNB,NULL
000066                                  TTAB$   STNW,NULL
000070                                  TTAB$   UNB,NULL
000072                                  TTAB$   UNW,NULL
000074                                  TTAB$   WORD,POP2
000076                                  TTAB$   APR,NULL
000100                                  TTAB$   DPRB,NULL
000102                                  TTAB$   DPRW,NULL
000104                                  TTAB$   TIME,NULL
000106                                  TTAB$   MXPTB,XMXPTB
000110                                  TTAB$   PECHA,NULL
000112                                  TTAB$   X3CH01,NULL
000114                                  TTAB$   X3CH02,NULL
000116                                  TTAB$   X3CH03,NULL
000120                                  TTAB$   X3CH04,NULL
000122                                  TTAB$   X3CH05,NULL
000124                                  TTAB$   X3CH06,NULL
000126                                  TTAB$   X3CH07,NULL
000130                                  TTAB$   X3CH08,NULL
000132                                  TTAB$   X3CH09,NULL
```

```
                                ;
                                ; LIBRARY POINTER
                                ;
001406                          LIBR:   .IF DF  M$$MGE

001406  112546                          MOVB    (R5)+,-(SP)     ; ASSEMBLE DESCRIPTOR BIAS
001410  112566  000001                  MOVB    (R5)+,1(SP)     ;       ...
001414  012600                          MOV     (SP)+,R0        ;       ...
001416  005024                          CLR     (R4)+           ; ZERO FIRST WORD OF ADDRESS (TEMPORARY)

                                        .IFTF

001420  112546                          MOVB    (R5)+,-(SP)     ; ASSEMBLE DESCRIPTOR ADDRESS
001422  112566  000001                  MOVB    (R5)+,1(SP)     ;       ...
001426  012603                          MOV     (SP)+,R3        ;       ...
001430  112524                          MOVB    (R5)+,(R4)+     ; COPY SECOND WORD OF ADDRESS
001432  112524                          MOVB    (R5)+,(R4)+     ;       ...

                                        .IFT

                                                .IF DF  R$$11M

001434                                  SWSTK$  10$             ;* ENTER KERNEL MODE

                                                .ENDC

001440                                  CALL    $MAPX           ;* MAP TO LIBRARY DESCRIPTOR
001444  016364  000000G 177774          MOV     LB.ADR(R3),-4(R4) ;* STORE LIBRARY BIAS
001452  016711  000000G                  MOV     $UBIAS,(R1)     ;* MAP TO UNLOAD BLOCK

                                        .IFF

                                        MOV     LB.ADR(R3),-2(R4) ; STORE LIBRARY ADDRESS
                                        MOV     R3,R0           ; CALCULATE START OF LIBRARY BLOCK
                                        BIC     #77,R0          ;       ...

                                        .ENDC

001456  016702  000000G                MOV     $LDA,R2         ;* GET LIBRARY DATA ADDRESS WITHIN UNLOAD BLOCK
001462  010022                          MOV     R0,(R2)+        ;* STORE DESCRIPTOR BIAS
001464  010312                          MOV     R3,(R2)         ;* AND OFFSET
001466  042722  177700                  BIC     #^C<77>,(R2)+   ;* 6 BITS ONLY
001472  010267  000000G                 MOV     R2,$LDA         ;* UPDATE LIBRARY DATA POINTER

001476                          10$:    RETURN
```

```
                        .TITLE  TMPEX3 - NTL TEMPLATE ERROR CLEANUP ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - LINE TABLE CLEANUP/DE-ALLOCATION ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 3.01  28-SEP-82
;       ADD .NODE, .PRTOV OPERATOR SUPPORT
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
;
;
;
;
```

```
                                ;+
                                ; $INT - ".INT" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       R0,R1,R2=DESTROYED
                                ;       $VECT1=CURRENT VALUE OF $SIZE
                                ;-
000032                          $INT::
000032                                  CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000036  012700  000000G                 MOV     #F1.INT,R0      ; BIT TO BE SET
000042  012701  000000C                 MOV     #F1.IN1!F1.IN2!F1.IN3,R1 ; BITS WHICH MUST NOT BE SET
000046  005002                          CLR     R2              ; VECTOR INDEX
000050                                  CALLR   VFVCT           ; CHECK VECTOR AVAILABILITY
```

```
                                        ;+
                                        ; $FILE - ".FILE" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       F2.FIL SET IF VALID FILE NAME SEEN ON THIS LINE
                                        ;       $FILEN=ASCIZ FILE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       SECONDARY TEMPLATE FILE PROCESSED
                                        ;-

                                                .ENABL  LSB

000000                                  $FILE::
000000  032767  00    000 000000G               BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN?
000006  001422                                  BEQ     10$             ; IF EQ, NO
000010  005000                                  CLR     R0              ; SETUP ASCIZ FILE SPEC
000012  012701  000000G                         MOV     #$FILEN,R1      ; FOR [131,5X]FILENAME.DAT
000016  012702  000000'                         MOV     #TMPEXT,R2      ;       ...
000022                                          CALL    $AZFS           ;       ...
000026                                          CALL    $CLOUE          ; TRY TO OPEN SECONDARY TEMPLATE
000032  105777  000000G                         TSTB    @$CLIOS         ; SUCCESS?
000036  100407                                  BMI     101$            ; IF MI, NO
000040  016746  000000G                         MOV     $VALUE,-(SP)    ; SAVE EXPRESSION VALUE
000044                                          CALLR   $TMFIX          ; PROCESS SECONDARY TEMPLATE
000050                                  $FILEX::                        ; OVERLAY RETURN ADDRESS
000050  012667  000000G                         MOV     (SP)+,$VALUE    ; RESTORE EXPRESSION VALUE
000054                                  10$:    RETURN

                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000056                                  101$:   EMSG$R  17              ; SECONDARY TEMPLATE NOT FOUND

                                                .DSABL  LSB
```

FILEID**TMPAC3

```
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC     333333
TTTTTTTTTT   MM      MM   PPPPPPPP       AAAAAA      CCCCCCCC     333333
    TT       MMMM  MMMM   PP      PP    AA    AA    CC            33    33
    TT       MMMM  MMMM   PP      PP    AA    AA    CC            33    33
    TT       MM  MM  MM   PP      PP    AA    AA    CC                  33
    TT       MM  MM  MM   PP      PP    AA    AA    CC                  33
    TT       MM      MM   PPPPPPPP     AA    AA    CC                33
    TT       MM      MM   PPPPPPPP     AA    AA    CC                33
    TT       MM      MM   PP           AAAAAAAAAA  CC                  33
    TT       MM      MM   PP           AAAAAAAAAA  CC                  33
    TT       MM      MM   PP           AA    AA    CC            33    33      ....
    TT       MM      MM   PP           AA    AA    CC            33    33      ....
    TT       MM      MM   PP           AA    AA      CCCCCCCC     333333       ....
    TT       MM      MM   PP           AA    AA      CCCCCCCC     333333       ....


LL             SSSSSSSS   TTTTTTTTTT
LL             SSSSSSSS   TTTTTTTTTT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL           SS               TT
LL             SSSSSS         TT
LL             SSSSSS         TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LL                   SS       TT
LLLLLLLLLL   SSSSSSSS         TT
LLLLLLLLLL   SSSSSSSS         TT
```

```
                                        .ENDC
                                        MOVB    $BFALL,(R2)+    ; AND COUNT (FOR CLEANUP)
000130  116722  000000G                 MOVB    $BFALL+1,(R2)   ;       ...
000134  116712  000001G
                                        .IF DF  M$$MGE
000140  062767  000004  000000G         ADD     #4,$SIZE        ; ADJUST DATA BASE SIZE
                                        .IFF
                                        ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
                                        .ENDC
000146  000300                          SWAB    R0              ; RESTORE ALLOCATION BIAS
000150  001424                          BEQ     30$             ; IF ZERO, MUX CHARACTERISRICS UPDATE
000152  016767  000000G 000000G         MOV     $BFLBN,$LBN     ; SET LOGICAL BLOCK NUMBER
000160  016767  000002G 000002G         MOV     $BFLBN+2,$LBN+2 ;       ...
000166  016767  000000G 000000G         MOV     $BFXFR,$LLEN    ; AND TRANSFER SIZE IN BYTES
000174                                  CALL    $READX          ; TRANSFER BINARY FILE INTO CORE
000200  103006                          BCC     25$             ; IF CC, NO PROBLEM
000202  116777  000000G 000000G         MOVB    $IOSB,@$CLIOS   ; COPY ERROR CODE
000210                                  EMSG$   77              ; READ FAILURE
000216                          25$:    CALL    $CLDEQ          ; CLOSE BINARY FILE
000222                          30$:    RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000224                          101$:   CALL    $CLDEQ          ; CLOSE BINARY FILE
000230                                  EMSG$R  26              ; RESOURCE ALLOCATION FAILURE
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  ASL$,EMSG$,EMSG$R,NHWDF$,NTLER$

                                ;***
                                ; LIBRARY SYMBOLS
                                ;***
000000                          NHWDF$

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000000                                  NTLER$  ,69,6,$TERR,$PRV.N,$EFNLL,<Open Failure (-***.)>
000036                                  NTLER$  ,71,6,$TERR,$REP.P,$EFNLL,<Not Contiguous>
000066                                  NTLER$  ,72,6,$TERR,$CUR.N,$EFNLL,<Label Block Read Failure (-***.)>
000140                                  NTLER$  ,73,6,$TERR,$REP.C,$EFNLL,<Not a Valid Binary Image>
                                        .IF DF  R$$11M
000202                                  NTLER$  ,74,6,$TERR,$REP.C,$EFNLL,<Size Exceeds 32K-32 Words>
                                        .IFF
                                        NTLER$  ,74,6,$TERR,$REP.C,$EFNLL,<Size Exceeds 4K Words>
                                        .ENDC
000244                                  NTLER$  ,75,6,$TERR,$REP.C,$EFNLL,<Extension Exceeds 32K Words>

                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRING
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000310  052     040     111    FMT6:    .ASCIZ  "* Image File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

                                ;
                                ; DEFAULT APR NUMBER
                                ;
                                        .IF DF  M$$MGE
        000006                  DFAPR=<<.BASEB/2>&77777>/10000
                                        .ENDC

000000                                  .PSECT  ..BUF

                                ;
                                ; DISK BUFFER FOR LABEL BLOCK READ
                                ;
000000                          DSKBUF: .BLKW   1
```

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  RESRG,SAVRG,EMSG$,NHWDF$,NTLER$
                                ;***
                                ; LIBRARY SYMBOLS
                                ;***
000000                                  NHWDF$
```

```
                                        .SBTTL  $X2CHB - .X2CHB ACTION ROUTINE
                                        .SBTTL  $X3CHB - .X3CHB ACTION ROUTINE

                                ;+
                                ;
                                ; $X2CHB - .X2CHB ACTION ROUTINE
                                ; $X3CHB - .X3CHB ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $VALUE - VALUE OF FIRST OPERAND (INDICATES WHICH PARAMETER
                                ;                FROM X2P$DF OR X3P$DF TO USE)
                                ;
                                ; OUTPUTS:
                                ;       R1 DESTROYED
                                ;
                                ;-
                                        .ENABL  LSB
001034  012701  000000G         $X2CHB::MOV     #.X2CH1,R1              ; GET FUNCTION CODE
001040  026727  000000G 000001          CMP     $VALUE,#X2BLO           ; IS VALUE WITHIN RANGE?
001046  103425                          BLO     10$                     ; BR IF NO
001050  026727  000000G 000003          CMP     $VALUE,#X2BHI           ; WITHIN RANGE?
001056  101021                          BHI     10$                     ; BR IF NO
001060  026727  000000G 000002          CMP     $VALUE,#X2BLK           ; BLOCK SIZE PARAMETER?
001066  001415                          BEQ     10$                     ; YES, NOT ALLOWED AS BYTE
001070                                  CALL    $VXLLC                  ; MAKE SURE IT'S NOT AN LLC
001074  000417                          BR      30$

001076  012701  000000G         $X3CHB::MOV     #.X3C01,R1              ; GET FUNCTION CODE
001102  026727  000000G 000003          CMP     $VALUE,#X3BLO           ; IS OPERAND WITHIN RANGE?
001110  103404                          BLO     10$                     ; BR IF NO
001112  026727  000000G 000013          CMP     $VALUE,#X3BHI           ; WITHIN RANGE?
001120  101403                          BLOS    20$                     ; BR IF YES
001122                          10$:    EMSG$   91                      ; INVALID OPERAND
001130                          20$:    CALL    $VFLLC                  ; MAKE SURE IT IS AN LLC
001134  066701  000000G         30$:    ADD     $VALUE,R1               ; POINT TO CORRECT FUNCTION CODE
001140  005301                          DEC     R1                      ; OPERAND STARTS AT 1
001142                                  CALL    $RQ1                    ; REQUEST BINARY BUFFER
001146  110122                          MOVB    R1,(R2)+                ; STORE FUNCTION CODE
001150  005267  000000G                 INC     $SIZE                   ; ADJUST DATA BASE SIZE
001154                                  RETURN
                                        .DSABL  LSB
```

```
                                ;+
                                ; MAX14 - ALLOW ONLY 14. ALLOCATIONS/LIBRARIES
                                ;
                                ; INPUTS:
                                ;       $NALL=NUMBER OF ALLOCATIONS
                                ;       $NLIB=NUMBER OF LIBRARIES
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;-
000252  116700  000000G       MAX14:  MOVB    $NALL,R0        ; GET NUMBER OF ALLOCATIONS
000256  116701  000000G               MOVB    $NLIB,R1        ; AND NUMBER OF LIBRARIES
000262  060001                        ADD     R0,R1           ; ADD THEM TOGETHER
000264  022701  000015                CMP     #13.,R1         ; ARE WE AT THE MAXIMUM ALREADY?
000270  103401                        BCS     101$            ; IF CS, YES
000272                                RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000274                        101$:   CALL    $LBERR          ; CLEANUP AFTER AN ERROR
000300                                EMSG$R  Y6              ; ALREADY 14. ALLOCS/LIBRS
```

```
                                ;+
                                ; $ALB - ALLOCATE LIBRARY BLOCK
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       $LBIAS, $LBADR=DESCRIPTOR ADDRESS
                                ;-
000000                          $ALB::
000000  012701  000001                  MOV     #BLKSIZ,R1      ; ALLOCATE 1 32. WORD BLOCK
000004                                  CALL    $ALN18          ; USE NON-UMR POOL SPACE
000010  103425                         BCS     101$            ; IF CS, FAILURE
000012  010067  000000G                 MOV     R0,LBLK         ; SAVE LIBRARY BLOCK ADDRESS

                                        .IF DF  M$$MGE

000016  010067  000000G                 MOV     R0,$LBIAS       ; SET DESCRIPTOR BIAS

                                        .IF DF  R$$11M

000022                                  SWSTK$  10$             ;* ENTER KERNEL MODE

                                        .ENDC

000026                                  CALL    $MAPX           ;* MAP TO THE LIBRARY BLOCK

                                        .IFF

                                        MOV     R0,R2           ; COPY LIBRARY BLOCK ADDRESS

                                        .ENDC

000032  005022                          CLR     (R2)+           ;* NO NEXT BLOCK IN LIST
000034  005022                          CLR     (R2)+           ;* NO DESCRIPTORS IN BLOCK YET
000036  010267  000000G                 MOV     R2,$LBADR       ;* SET DESCRIPTOR ADDRESS
000042  005062  000014                  CLR     12.(R2)         ;* INDICATE NO OTHER LIBRARIES
000046  005062  000030                  CLR     24.(R2)         ;*      ...
000052  005062  000044                  CLR     36.(R2)         ;*      ...
000056  005062  000060                  CLR     48.(R2)         ;*      ...

                                        .IF DF  R$$11D ! I$$AS
                                        CALL    @(SP)+          ;* CALL CO-ROUTINE TO RESTORE MAPPING
                                        .ENDC

000062                          10$:    RETURN                  ;*

                                ;
                                ; ERROR CONDITION
                                ;
000064                          101$:   CALL    $LBERR          ; CLEANUP AFTER AN ERROR
000070                                  EMSG$R  Y1              ; LIBRARY BLOCK ALLOCATION FAILURE

                                        .END
```

Table of contents

5-	54	MACRO DEFINITIONS
6-	71	LOCAL DATA
7-	97	ERROR MESSAGES
8-	149	$TMALT - ALLOCATE LINE TABLE
9-	306	KRBAL - ALLOCATE THE KRB
10-	431	$TMCHK - CHECK FOR CORRECT DDM INTERRUPT LINKAGE
11-	478	UBA - GET UNLOAD BLOCK ADDRESS
12-	640	GETIND - GET PDV INDEX
13-	670	UBLTA - STORE LINE TABLE ADDRESS
14-	696	XLT - DEALLOCATE LINE TABLE
14-	697	XICB - DEALLOCATE ICB
15-	785	.DEA16 - END OF PROCESS DEALLOCATION
16-	882	$ALICB - ALLOCATE AN ICB
17-	1059	ALOC2 - ALLOCATE FROM OUTSIDE TASK SPACE
18-	1116	DEAC2 - DEALLOCATE TO OUTSIDE TASK SPACE

```
     535 000732  032723  000001                BIT     #1,(R3)+        ;* ANY ALLOCATIONS THIS ENTRY?
     536 000736  001770                        BEQ     30$             ;* IF EQ, NO
     537 000740  112304                        MOVB    (R3)+,R4        ;* GET ALLOCATION COUNT
     538 000742  112305                        MOVB    (R3)+,R5        ;* AND LIBRARY COUNT
     539 000744  060405                        ADD     R4,R5           ;* ADD THEM TOGETHER
     540 000746  006305                        ASL     R5              ;* MULTIPLY BY 4 (2 WORDS PER ENTRY)
     541 000750  006305                        ASL     R5              ;*
     542 000752  060503                        ADD     R5,R3           ;* SKIP THE ALLOC/LIBR ENTRIES
     543 000754  000761                        BR      30$             ;*
     544 000756  024343              40$:      CMP     -(R3),-(R3)     ;* POINT BACK AT THE ZERO WORD
     545 000760  010304                        MOV     R3,R4           ;* COPY VIRTUAL ADDRESS
     546 000762  066704  000000G               ADD     $LDA,R4         ;* INCLUDE UNLOAD SIZE NEEDED
     547 000766  020400                        CMP     R4,R0           ;* IS THERE ENOUGH SPACE IN THIS BLOCK?
     548 000770  101003                        BHI     50$             ;* IF HI, NO
     549 000772                                CALL    UBLTA           ;* STORE LINE TABLE ADDRESS IN UNLOAD BLOCK
     550 000776  000402                        BR      60$             ;*
     551
     552                                       .IFT
     553
     554 001000  011211              50$:      MOV     (R2),(R1)       ;* MAP TO NEXT UNLOAD BLOCK
     555 001002  001341                        BNE     20$             ;* IF NE, NOT AT END YET
     556
     557                                       .IF DF  R$$11M
     558
     559 001004                      60$:      RETURN                  ;* BACK TO USER MODE
     560
     561                                       .IFF
     562
     563                             60$:      CALL    @(SP)+          ;;; RESTORE MAPPING AND RE-ENABLE TASK SWITCHING
     564
     565                                       .ENDC
     566
     567                                       .IFF
     568
     569                             50$:      MOV     (R2),R2         ; GET NEXT UNLOAD BLOCK ADDRESS
     570                                       BNE     20$             ; IF NE, NOT AT END YET
     571                             60$:
     572
     573                                       .IFTF
     574
     575 001006  005767  000000G     70$:      TST     $UDA            ; HAS ADDRESS BEEN CALCULATED?
     576 001012  001041                        BNE     100$            ; IF NE, YES
     577
     578                                       .IFT
     579
     580 001014  012701  000001                MOV     #BLKSIZ,R1      ; ALLOCATE 1 32. WORD BLOCK
     581 001020                                CALL    $ALN18          ; USE NON-UMR POOL SPACE
     582 001024  103435                        BCS     101$            ; IF CS, FAILURE
     583
     584                                       .IF DF  R$$11M
     585
     586 001026                                SWSTK$  100$            ;* ENTER KERNEL MODE
     587
     588                                       .ENDC
     589
     590 001032                                CALL    $MAPX           ;* MAP TO JUST ALLOCATED BLOCK
     591 001036  010203                        MOV     R2,R3           ;* COPY VIRTUAL ADDRESS
```

```
                        .SBTTL  DEAC2 - DEALLOCATE TO OUTSIDE TASK SPACE
                ;+
                ; *** - DEAC2 - DEALLOCATE POOL SPACE TO AREA OUTSIDE TASK'S SPACE
                ;
                ; INPUTS:
                ;       R0 = ADDRESS OF THE CORE BUFFER TO BE DE-ALLOCATED
                ;       R1 = LENGTH OF THE CORE BUFFER
                ;       R2 = DISPLACEMENT ADDRESS (100000 OR 120000)
                ;       R3 = ADDRESS OF CORE ALLOCATION LISTHEAD-2
                ;
                ; OUTPUTS:
                ;       R0-R3 = DESTROYED
                ;
                ;       THE CORE BLOCK IS MERGED INTO THE FREE CORE CHAIN BY CORE
                ;       ADDRESS AND IS AGLOMERATED IF NECESSARY WITH ADJACENT BLOCKS.
                ;
                ;-
                        .IF DF  R$$MPL
                DEAC2:
                        ADD     (R3),R1         ; ROUND TO NEXT BOUNDARY
                        BIC     (R3)+,R1        ; CLEAR EXCESS
                        BEQ     60$             ; IF EQ, NO BLOCK TO RELEASE
                        MOV     R2,-(SP)        ; SAVE DISPLACEMENT
                20$:    MOV     R3,R2           ; SAVE ADDRESS OF CURRENT BLOCK
                        MOV     (R2),R3         ; GET ADDRESS OF NEXT BLOCK
                        BEQ     30$             ; IF EQ, END OF CHAIN
                        BIC     #160000,R3      ; CLEAR CURRENT MAPPING DISPLACEMENT
                        BIS     (SP),R3         ; USE OUR MAPPING DISPLACEMENT
                        CMP     R0,R3           ; BLOCK GO HERE ?
                        BHIS    20$             ; IF HIS, NO
                30$:    MOV     R3,(R0)         ; ASSUME NO AGLOMERATION
                        MOV     R0,-(SP)        ; CALCULATE ADDRESS OF NEW BLOCK
                        ADD     R1,(SP)         ;
                        MOV     2(SP),-(SP)     ;; CALCULATE END OF APR ADDRESSING
                        BIS     #17777,(SP)     ;;
                        CMP     (SP)+,(SP)      ;; BIGGER THAN APR ADDRESSING RANGE ?
                        BLO     80$             ;; IF LO, YES
                        CMP     R3,(SP)+        ; EQUAL TO NEXT IN CHAIN ?
                        BLO     70$             ; IF LO, DE-ALLOCATION OVERLAPS
                        BNE     40$             ; IF NE, NO
                        MOV     (R3)+,(R0)      ; MOVE LINK WORD TO BLOCK RELEASED
                        ADD     (R3),R1         ; MERGE TWO BLOCKS
                40$:    MOV     R2,-(SP)        ; SAVE ADDRESS OF PREVIOUS BLOCK
                        MOV     R0,(R2)         ; ASSUME NO AGLOMERATION
                        ADD     (R2),(SP)       ; CALCULATE ADDRESS OF NEXT BLOCK
                        CMP     R0,(SP)+        ; EQUAL TO BLOCK BEING RELEASED ?
                        BNE     50$             ; IF NE, NO
                        ADD     (R2),R1         ; MERGE TWO BLOCKS
                        MOV     (R0)+,-(R2)     ; MOVE LINK WORD TO PREVIOUS BLOCK
                        MOV     R2,R0           ; SET NEW ADDRESS OF BLOCK
                50$:    MOV     R1,2(R0)        ; SET SIZE OF BLOCK RELEASED
                        MOV     (SP)+,R2        ; RESTORE ADDRESS OF LISTHEAD
                60$:    RETURN

                70$:    TST     R3              ;; INSERT AT END OF LISTH ?
                        BEQ     40$             ;; IF EQ, YES - OKAY
```

TMPERR CREATED BY MACRO ON 29-JUN-85 AT 00:42 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES									
EMSG$R	#5-59 6-119	6-110	6-111	6-112	6-113	6-114	6-115	6-116	6-117	6-118
NTLER$	#5-59 5-84	5-71	5-72	5-73	5-74	5-79	5-80	5-81	5-82	5-83

FILEID**TMPEXP

LOCAL DATA

```
    135 000134                          TTAB$   X3CH10,NULL
    136 000136                          TTAB$   X3CH11,NULL
    137 000140                          TTAB$   X2CH01,NULL
    138 000142                          TTAB$   X2CH02,NULL
    139 000144                          TTAB$   X2CH03,NULL
    140 000146                          TTAB$   X2CH04,NULL
    141 000150                          TTAB$   X2CH05,NULL
    142
    143                                 ;
    144                                 ; SECONDARY FILE INDEX VALUE
    145                                 ;
    146 000152                          INDEX:  .BLKW   1
    147
    148                                 ;
    149                                 ; MUX/MTP FLAGS
    150                                 ;
    151 000154                          FLAGS:  .BLKW   1               ; 100000=MUX, 200=MTP
    152
    153 000000                                  .PSECT
```

```
                                        ;
                                        ; SECONDARY MUX FILE ADDRESS
                                        ;
001500  012767  100000  000154' MXPTB:  MOV     #100000,FLAGS           ; INDICATE MUX EXPANSION
001506  112546                         MOVB    (R5)+,-(SP)             ; GET REPEAT COUNT
001510  112524                         MOVB    (R5)+,(R4)+             ; GET MUX TABLE ADDRESS
001512  112524                         MOVB    (R5)+,(R4)+             ; ...
001514  010446                         MOV     R4,-(SP)                ; SAVE OUTPUT FILE POINTER
001516  016404  177776                 MOV     -2(R4),R4               ; GET NEW OUTPUT FILE POINTER
001522  010546                         MOV     R5,-(SP)                ; SAVE CURRENT BINARY FILE POINTER
001524  011605                 10$:    MOV     (SP),R5                 ; RESTORE BINARY FILE POINTER
001526  005003                 20$:    CLR     R3                      ; GET FUNCTION CODE
001530  152503                         BISB    (R5)+,R3                ; ...
001532  122703  000000G                CMPB    #.EOF,R3                ; IS IT END OF FILE ?
001536  001404                         BEQ     30$                     ; IF EQ, YES
001540  106303                         ASLB    R3                      ; CONVERT TO A WORD INDEX
001542                                  CALL    @TABLE-2(R3)            ; EXPAND A FIELD
001546  000767                         BR      20$                     ; GO TILL END OF FILE
001550  005267  000152'         30$:    INC     INDEX                   ; TALLY ONE PASS THRU SECONDARY FILE
001554  126667  000004  000152'        CMPB    4(SP),INDEX             ; ARE WE DONE ?
001562  001360                         BNE     10$                     ; IF NE, NO
001564  005726                         TST     (SP)+                   ; PURGE BINARY FILE POINTER
001566  012604                         MOV     (SP)+,R4                ; RESTORE OUTPUT FILE POINTER
001570  005726                         TST     (SP)+                   ; PURGE COUNT
001572  005067  000152'                CLR     INDEX                   ; CLEAR SECONDARY INDEX
001576  005067  000154'                CLR     FLAGS                   ; AND SPECIAL FLAGS
001602  016703  000000G                MOV     $MXSIZ,R3               ; GET SIZE OF MUX TABLES
001606                                  CALLR   SAVALC                  ; SAVE ADDRESS AND SIZE IN UNLOAD BLOCK


                                        ;
                                        ; CURRENT TIME
                                        ;
001612  017746  000000G         TIME:   MOV     @ZTIME,-(SP)            ; GET TIME SINCE MIDNIGHT       ;LAA3.01
001616  111624                         MOVB    (SP),(R4)+              ; STORE TIME                    ;LAA3.01
001620  116624  000001                 MOVB    1(SP),(R4)+             ;                               ;LAA3.01
001624  005726                         TST     (SP)+                   ; CLEAN UP STACK                ;LAA3.01
001626                                  RETURN                          ;                               ;LAA3.01
                                                                        ;                               ;**-4


                                        ;
                                        ; ACTIVE POLLING RATIO (APR)
                                        ;
001630  105767  000154'         APR:    TSTB    FLAGS                   ; IS IT MTP EXPANSION ?
001634  100005                         BPL     101$                    ; IF PL, NO
001636  016700  000152'                 MOV     INDEX,R0                ; GET CURRENT STATION INDEX
001642  116024  000000G                MOVB    $$APR(R0),(R4)+         ; STORE APR
001646                                  RETURN
001650  112724  177777         101$:   MOVB    #-1,(R4)+               ; APR NOT IN MPTAB FILE
001654                                  RETURN


                                        ;
                                        ; DEAD POLLING RATIO BYTE
                                        ;
001656  116724  000000G         DPRB:   MOVB    $$DPR,(R4)+             ; STORE DPR
001662                                  RETURN
```

```
     58
```

```
                                        ;+
                                        ; $INT1 - ".INT1" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $VECT1=CURRENT COPY OF $SIZE
                                        ;-
000054                                  $INT1::
000054                                          CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000060  012700  000000G                         MOV     #F1.IN1,R0      ; BIT TO BE SET
000064  012701  000000G                         MOV     #F1.INT,R1      ; BIT WHICH MUST NOT BE SET
000070  005002                                  CLR     R2              ; VECTOR INDEX
000072                                          CALLR   VFVCT           ; CHECK VECTOR AVAILABILITY
```

```
                                        ;+
                                        ; $MXPTB - ".MXPTB" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $$MUX/$VALUE=FILE REPEAT COUNT
                                        ;       $FILEN=SECONDARY TEMPLATE FILE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       $MXADD=MUX TABLE ADDRESS
                                        ;       $MXLEN=SIZE OF SECONDARY TEMPLATE FILE
                                        ;       $MXSIZ=SIZE OF ALL MUX TABLES
                                        ;       R0,R1,R2=DESTROYED
                                        ;
                                        ;-
                                        $MXPTB::
000064                                          CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000064                                          CALL    $VFMUX          ; AND A MUX LINE
000070
000074  012700  000000G                        MOV     #F1.MXT,R0      ; MAKE SURE THIS IS THE ONLY OCCURANCE
000100                                          CALL    $VFSPC          ; ...
000104  016767  000000G 000000G                 MOV     $$MUX,$VALUE    ; DO IT <UNIT COUNT> TIMES

000112  022767  000400  000000G                 CMP     #256.,$VALUE    ; IS COUNT > 256. ?
000120  103500                                  BLO     101$            ; IF LO, YES
000122  005767  000000G                        TST     $VALUE          ; IS COUNT 0 ?
000126  001474                                  BEQ     30$             ; IF EQ, YES
000130                                          CALL    MAX14           ; ALLOW ONLY 14. ALLOCATIONS
000134  103471                                  BCS     30$             ; IF CS, NO MORE ALLOWED
000136                                          CALL    $RQ4            ; REQUEST BINARY FILE
000142  103466                                  BCS     30$             ; IF CS, NO ACCESS
000144  112722  000000G                        MOVB    #.MXPTB,(R2)+   ; STORE FUNCTION CODE
000150  116722  000000G                        MOVB    $VALUE,(R2)+    ; AND FILE REPEAT COUNT
000154  010246                                  MOV     R2,-(SP)        ; SAVE POINTER TO BINARY INFO.
000156  016746  000000G                        MOV     $SIZE,-(SP)     ; SAVE CURRENT LINE TABLE SIZE
000162                                          CALL    $FILE           ; PROCESS SECONDARY TEMPLATE
000166                                          CALL    $EVEN           ; AUTOMATIC WORD RE-ALIGNMENT
000172  016700  000000G                        MOV     $SIZE,R0        ; SAVE CURRENT POSITION IN BINARY BUFFER
000176  012667  000000G                        MOV     (SP)+,$SIZE     ; RESTORE SIZE FROM START OF FILE
000202  166700  000000G                        SUB     $SIZE,R0        ; CALCULATE SIZE OF FILE
000206  012602                                  MOV     (SP)+,R2        ; RESTORE POINTER TO BINARY BUFFER
000210  005001                                  CLR     R1              ; INITIALIZE DATA BASE SIZE
000212  060001                          10$:    ADD     R0,R1           ; CALCULATE DATA BASE SIZE
000214  005367  000000G                        DEC     $VALUE          ; BY <FILE COUNT> * <FILE SIZE>
000220  001374                                  BNE     10$             ; ...
000222  005400                                  NEG     R0              ; GET FILE SIZE AS NEGATIVE LENGTH
000224  010067  000000G                        MOV     R0,$MXLEN       ; SAVE MUX TABLE LENGTH
000230  005000                                  CLR     R0              ; ASSUME MUX UPDATE
000232  032767  000000G 000000G                 BIT     #FL.CHA,$FLAGS  ; MUX UPDATE ?
000240  001014                                  BNE     20$             ; IF NE, YES
000242  005201                                  INC     R1              ; ROUND UP TO MULTIPLE OF 2
000244  042701  000001                        BIC     #1,R1           ; ...
000250  010167  000000G                        MOV     R1,$MXSIZ       ; SAVE MUX TABLE SIZE
000254                                          CALL    $ALL16          ; TRY TO ALLOCATE MUX TABLES
000260  103423                                  BCS     111$            ; IF CS, ALLOCATION ERROR
000262  105267  000000G                        INCB    $NALL           ; TALLY SUCCESSFUL ALLOCATION
000266  010067  000000G                        MOV     R0,$MXADD       ; SAVE ADDRESS OF 'TMPVCT'
000272  110022                          20$:    MOVB    R0,(R2)+        ; STORE ADDRESS
```

```
                        .TITLE  TMPAC3 - NTL TEMPLATE PROCESSING ACTION ROUTINES
                        .IDENT  /V05.00/


                ;
                ; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
                ; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                ;
                ;
                ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
                ; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
                ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
                ; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
                ; TRANSFERRED.
                ;
                ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
                ; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
                ; CORPORATION.
                ;
                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
                ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                ;
                ; MODULE DESCRIPTION:
                ;
                ;       NTL - TEMPLATE SYNTAX TPARS ACTION ROUTINES, PART 3
                ;
                ; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
                ;
                ; IDENT HISTORY:
                ;
                ; 1.00  27-FEB-78
                ;       VERSION 2.0 RELEASE
                ;
                ; 2.00  14-DEC-79
                ;       DECNET-11M/S V3.0
                ;       DECNET-11M-PLUS V1.0
                ;
                ; 3.00  16-APR-82
                ;       DECNET-11M V3.1
                ;       DECNET-11M-PLUS V1.1
                ;
                ; 4.00  07-NOV-83
                ;       DECNET-11M V4.0
                ;       DECNET-11M-PLUS V2.0
                ;
                ; 5.00  22-JUL-85
                ;       DECnet-11M/S V4.2
                ;       DECnet-11M-Plus V3.0
                ;       DECnet-Micro/RSX V1.0
                ;
                ;
```

```
                                        ;+
                                        ; $CORE1 - ".CORE" OPERAND 1 ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=NUMBER OF BYTES NEEDED
                                        ;
                                        ; OUTPUTS:
                                        ;       $CVAL=NUMBER OF BYTES NEEDED
                                        ;       $VALUE=DEFAULT MAPPING DISPLACEMENT
                                        ;       $CAPR=APR NUMBER
                                        ;-
000236                                  $CORE1::
                                                .IF DF  M$$MGE

000236  016700  000000G                        MOV     $VALUE,R0       ; GET EXPRESSION VALUE
000242  062700  000077                         ADD     #77,R0          ; ROUND UP TO NEXT MULTIPLE OF 100
000246  006000                                 ROR     R0              ;       ...
000250  000241                                 CLC                     ;       ...
000252  006000                                 ROR     R0              ;       ...
000254                                         ASR$    4,R0            ;       ...
000264  010067  000000G                        MOV     R0,$CVAL        ; SET NEW EXPRESSION VALUE
000270  012767  140000  000000G               MOV     #.BASE8,$VALUE  ; SET DEFAULT MAPPING
000276  112767  000006  000000G               MOVB    #DFAPR,$CAPR    ; SET DEFAULT APR NUMBER

                                                .IFF

                                                MOV     $VALUE,$CVAL    ; COPY EXPRESSION VALUE

                                                .ENDC

000304                                          RETURN
```

```
    113
    114 000000                          .PSECT
```

```
                                .SBTTL  CONSTANT DEFINITIONS

000001          X3WLO = 1               ; LOW VALUE FOR .X3CHW OPERATOR
000002          X3WHI = 2               ; HIGH VALUE FOR .X3CHW OPERATOR
000003          X3BLO = 3               ; LOW VALUE FOR .X3CHB OPERATOR
000013          X3BHI = 11.             ; HIGH VALUE FOR .X3CHB OPERATOR
000004          X2WLO = 4               ; LOW VALUE FOR .X2CHW OPERATOR
000005          X2WHI = 5               ; HIGH VALUE FOR .X2CHW OPERATOR
000001          X2BLO = 1               ; LOW VALUE FOR .X2CHB OPERATOR
000003          X2BHI = 3               ; HIGH VALUE FOR .X2CHB OPERATOR
000002          X2BLK = 2               ; BLOCK SIZE - .X2CHW SPECIAL CASE (SORRY)
```

```
                                        .SBTTL  $VFLLC - VERIFY LLC

                                ;+
                                ; $VFLLC - VERIFY LLC ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       ERROR UNLESS LLC BEING LOADED
                                ;-
001156                          $VFLLC:
001156  032767  000000G 000000G         BIT     #FL.LLC,$FLAGS  ; IS THIS AN LLC?
001164  001003                          BNE     10$             ; IF NE, YES
001166                                  EMSG$   89              ; LLC OPERATOR IN DDM/DLC TEMPLATE
001174                          10$:    RETURN
```

```
                                ;+
                                ; LINK - LINK NEWLY ALLOCATED LIBRARY BLOCK INTO LIST
                                ;
                                ; INPUTS:
                                ;       LBLK=LIBRARY BLOCK ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
000306                          LINK:
                                
000306  017700  000000G                 MOV     @NTLPT,R0       ; GET NETLDR DATA BLOCK BIAS
000312                                  SWSTK$  30$             ;* ENTER KERNEL MODE
000316                                  CALL    $MAPX           ;* MAP TO THE DATA BLOCK
000322  016211  000000G                 MOV     .CXLBR(R2),(R1) ;* AND TO THE FIRST LIBRARY BLOCK
000326  001006                          BNE     20$             ;* IF NE, THERE IS AT LEAST ONE
000330  010011                          MOV     R0,(R1)         ;* RE-MAP TO THE DATA BLOCK
000332  016762  000000' 000000G         MOV     LBLK,.CXLBR(R2) ;* SET FIRST LIBRARY BLOCK ADDRESS
000340  000405                          BR      30$             ;*
000342  011211                  10$:    MOV     (R2),(R1)       ;* MAP TO THE NEXT LIBRARY BLOCK
                                
000344  005712                  20$:    TST     (R2)            ;* IS THIS THE LAST BLOCK IN LIST?
000346  001375                          BNE     10$             ;* IF NE, NO
000350  016712  000000'                 MOV     LBLK,(R2)       ;* SET LAST LIBRARY BLOCK ADDRESS
                                
000354                          30$:    RETURN                  ;*
```

```
TMPAC8 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:41  Page 6-1
Symbol table

A$$CHK= 000000          FMT2  = ****** GX        LD$LP = 000000          N$$MCP= 000001          T$$KMG= 000000
A$$CPS= 000000          FM.2  = 000000          L$$ASG= 000000          N$$MLL= 000001          T$$MIN= 000000
A$$PRI= 000000          F$$LVL= 000001          L$$DRV= 000000          N$$MOV= 000010          V$$CTR= 001000
A$$TRP= 000000          G$$TPP= 000000          L$$P11= 000001          N$$NCT= 000001          X$$DBT= 000000
BLKSIZ= 000001          G$$TSS= 000000          L$$11R= 000000          N$$PEM= 000001          $ALB    000000RG
C$$CKP= 000000          G$$TTK= 000000          M$$CRB= 000124          P$$P45= 000000          $ALN18= ****** CX
C$$ORE= 000400          G$$WRD= 000000          M$$CRX= 000000          P$$WRD= 000000          $ELINE= ****** GX
C$$RSH= 177564          I$$RAR= 000000          M$$FCS= 000000          Q$$OPT= 000010          $ERRY1  000000R     002
D$$BUG= 177514          I$$RDN= 000000          M$$MGE= 000000          R$$DER= 000000          $LBADR= ****** GX
D$$ISK= 000000          K$$CNT= 177546          M$$NET= 000000          R$$K11= 000001          $LBERR= ****** GX
D$$L11= 000001          K$$CSR= 177546          M$$OVR= 000000          R$$SND= 000000          $LBIAS= ****** GX
D$$YNC= 000000          K$$LDC= 000000          N$$ACC= 000001          R$$11M= 000000          $MAPX = ****** GX
D$$YNM= 000000          K$$TPS= 000074          N$$BUF= 000001          S$$WRG= 000000          $REP.C= ****** GX
E$$XPR= 000000          LBLK  = ****** GX       N$$LDV= 000001          S$$YSZ= 007600          $TERR = ****** GX


. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        000076    001   (RW,I,LCL,REL,CON)
DATA    000052    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 10149 Words  ( 40 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:06.43
SY:TMPAC8.V2,[132,134]TMPAC8/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC8
```



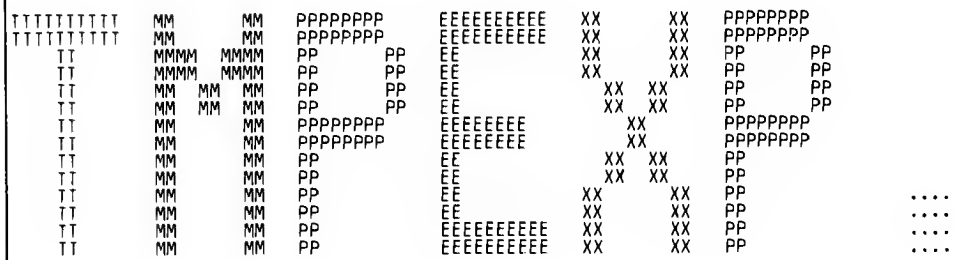
```
TMPAC8     CREATED BY  MACRO  ON 29-JUN-85 AT 00:41     PAGE 1
```

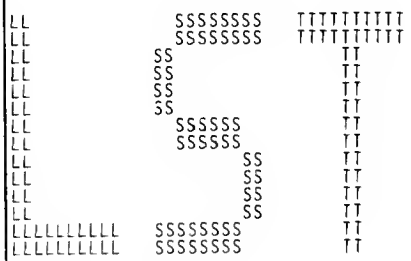
```
                        .TITLE  TMPALT - TEMPLATE ALLOCATION ROUTINES
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - LINE TABLE ALLOCATION ROUTINES
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
```

```
 592 001040  005023                        CLR     (R3)+           ;* CLEAR LINK TO NEXT UNLOAD BLOCK
 593 001042  010304                        MOV     R3,R4           ;* ZERO THE REMAINDER OF THE 32. WORD BLOCK
 594 001044  012705  000037                MOV     #31.,R5         ;*      ...
 595 001050  005024                 75$:   CLR     (R4)+           ;*      ...
 596 001052                                SOB     R5,75$          ;*      ...
 597 001056                                CALL    UBLTA           ;* STORE LINE TABLE ADDRESS DATA
 598 001062  016711  000604'               MOV     NLBLK,(R1)      ;* MAP TO NETLDR DATA BLOCK
 599 001066  016211  000000G               MOV     .CXUNL(R2),(R1) ;* SET FIRST UNLOAD BLOCK ADDRESS
 600 001072  001006                        BNE     90$             ;* IF NE, THERE'S AT LEAST ONE
 601 001074  016711  000604'               MOV     NLBLK,(R1)      ;* RE-MAP TO NETLDR DATA BLOCK
 602 001100  010062  000000G               MOV     R0,.CXUNL(R2)   ;* SET FIRST UNLOAD BLOCK ADDRESS
 603 001104  000404                        BR      100$            ;*
 604 001106  011211                 80$:   MOV     (R2),(R1)       ;* MAP TO NEXT UNLOAD BLOCK
 605
 606                                       .IFF
 607
 608                                       MOV     #BLKSIZ,R1      ; ALLOCATE 1 32. WORD BLOCK
 609                                       CALL    $ALLTB          ;      ...
 610                                       BCS     101$            ; IF CS, FAILURE
 611                                       MOV     R0,R3           ; COPY UNLOAD BLOCK ADDRESS
 612                                       CLR     (R3)+           ; CLEAR LINK TO NEXT UNLOAD BLOCK
 613                                       MOV     R3,R4           ; ZERO THE REMAINDER OF THE 32. WORD BLOCK
 614                                       MOV     #31.,R5         ;      ...
 615                                75$:   CLR     (R4)+           ;      ...
 616                                       SOB     R5,75$          ;      ...
 617                                       CALL    UBLTA           ; STORE LINE TABLE ADDRESS DATA
 618                                       MOV     @NTLPT,R1       ; GET NETLDR DATA BLOCK ADDRESS
 619                                       MOV     .CXUNL(R1),R2   ; GET FIRST UNLOAD BLOCK ADDRESS
 620                                       BNE     90$             ; IF NE, THERE'S AT LEAST ONE
 621                                       MOV     R0,.CXUNL(R1)   ; SET FIRST UNLOAD BLOCK ADDRESS
 622                                       BR      100$            ;
 623                                80$:   MOV     (R2),R2         ; GET NEXT UNLOAD BLOCK ADDRESS
 624
 625                                       .ENDC
 626
 627 001110  005712                 90$:   TST     (R2)            ;* IS THIS THE LAST UNLOAD BLOCK?
 628 001112  001375                        BNE     80$             ;* IF NE, NO
 629 001114  010012                        MOV     R0,(R2)         ;* SET LAST DATA BLOCK ADDRESS
 630
 631 001116                         100$:  RETURN
 632
 633                                       ;
 634                                       ; ERROR CONDITION
 635                                       ;
 636 001120                         101$:  CALL    XICB            ; RELEASE ICB
 637 001124                                CALL    XLT             ; RELEASE LINE TABLE
 638 001130                                EMSG$R  86              ; UNLOAD BLOCK ALLOCATION FAILURE
```

```
                                        80$:    CRASH                   ;; ILLEGAL DE-ALLOCATION
                                                .ENDC   ; DF R$$MPL
                                        
                                        
                                        
000001                                          .END
```

```
**FILE**ID**TMPEXP
```





```
                                        .SBTTL  $TMEX2 - EXPAND BINARY INFO


                                ;+
                                ; $TMEX2 - EXPAND BINARY INFORMATION INTO REAL INFORMATION
                                ;
                                ; THIS ROUTINE IS CALLED AFTER THE ASCII TEMPLATE FILE HAS BEEN READ
                                ; AND PROCESSED INTO THE BINARY FILE. THE BINARY INFORMATION IS EXPANDED
                                ; AND STORED AT THE LOCATION CONTAINED IN "$ADDR".
                                ;
                                ; INPUTS:
                                ;       $BIN=START OF BINARY INFORMATION
                                ;       $PTR=END OF BINARY INFORMATION
                                ;       $ADDR=ADDRESS OF WHERE REAL DATA GOES
                                ;
                                ; OUTPUTS:
                                ;       ALL REGISTERS=DESTROYED
                                ;-
000000                          $TMEX2::
000000  005067  000152'                CLR     INDEX           ; ZERO INDEX VALUE
000004  005067  000154'                CLR     FLAGS           ; AND SPECIAL FLAGS
000010  016705  000000G                MOV     $BIN,R5         ; START OF BINARY DATA
000014  016704  000000G                MOV     $ADDR,R4        ; START OF REAL DATA
000020  026705  000000G        10$:    CMP     $PTR,R5         ; IS IT END OF BINARY DATA?
000024  001407                         BEQ     20$             ; IF EQ, YES
000026  005003                         CLR     R3              ; GET FUNCTION CODE
000030  152503                         BISB    (R5)+,R3        ;  ...
000032  001404                         BEQ     20$             ; IF ZERO, END OF FILE
000034  106303                         ASLB    R3              ; CONVERT TO A WORD INDEX
000036                                 CALL    @TABLE-2(R3)    ; EXPAND ONE FIELD
000042  000766                         BR      10$             ; KEEP LOOPING UNTIL END
000044                          20$:    RETURN
```

```
                                        ;
                                        ; DEAD POLLING RATIO WORD
                                        ;
001664  016724  000000G         DPR::   MOV     $$DPR,(R4)+             ; STORE DPR
001670                                  RETURN
```

```
                                ;***
                                ; LOCAL MACROS
                                ;***
                                ;
                                ; DEFINE TRANSFER TABLE ENTRY
                                ;
                                        .MACRO  TTAB$   AA,BB
                                        .WORD   BB
                                        .ENDM   TTAB$

                                ;***
                                ; LOCAL DATA
                                ;***
000000                                  .PSECT  DATA,D

                                ;
                                ; TRANSFER TABLE USING FUNCTION CODE AS INDEX
                                ;
000000                          TABLE:  TTAB$   ADDRB,NULL      ; EXPANSION ROUTINE IS FIRST ENTRY
000002                                  TTAB$   ADDRW,NULL      ; CLEANUP ROUTINE IS SECOND ENTRY
000004                                  TTAB$   BLKB,POP1
000006                                  TTAB$   BLKW,POP1
000010                                  TTAB$   BYTE,POP1
000012                                  TTAB$   CORE,XCORE
000014                                  TTAB$   CSR,POP1
000016                                  TTAB$   CTIM,NULL
000020                                  TTAB$   DVCHA,NULL
000022                                  TTAB$   EOF,NULL
000024                                  TTAB$   FILE,POP1
000026                                  TTAB$   INT,POP2
000030                                  TTAB$   LIBR,XLIBR
000032                                  TTAB$   LINKS,NULL
000034                                  TTAB$   LFILE,POP3
000036                                  TTAB$   LSTHD,NULL
000040                                  TTAB$   LTAB,NULL
000042                                  TTAB$   MPTAB,POP3
000044                                  TTAB$   MXTAB,POP1
000046                                  TTAB$   NOD,NULL
000050                                  TTAB$   PRI,NULL
000052                                  TTAB$   PRIOV,POP1
000054                                  TTAB$   SCOM,XSCOM
000056                                  TTAB$   SECSR,NULL
000060                                  TTAB$   SLNB,NULL
000062                                  TTAB$   SLNW,NULL
000064                                  TTAB$   STNB,NULL
000066                                  TTAB$   STNW,NULL
000070                                  TTAB$   UNB,NULL
000072                                  TTAB$   UNW,NULL
000074                                  TTAB$   WORD,POP2
000076                                  TTAB$   APR,POP1
000100                                  TTAB$   DPRB,POP1
000102                                  TTAB$   DPRW,POP2
000104                                  TTAB$   TIME,POP2
000106                                  TTAB$   MXPTB,XMXPTB
000110                                  TTAB$   PECHA,NULL
```

```
                                        ;+
                                        ; $INT2 - ".INT2" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $VECT2=CURRENT COPY OF $SIZE
                                        ;-
000076                                  $INT2::
000076                                          CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000102  012700  000000G                         MOV     #F1.IN..,R0     ; BIT TO BE SET
000106  012701  000000G                         MOV     #F1.INT,R1      ; BIT WHICH MUST NOT BE SET
000112  012702  000002                          MOV     #2,R2           ; VECTOR INDEX
000116                                          CALLR   VFVCT           ; CHECK VECTOR AVAILABILITY
```

```
    200 000274  000300                          SWAB    R0              ; ...
    201 000276  110022                          MOVB    R0,(R2)+        ; ...
    202 000300  062767  000002  000000G         ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
    203 000306                                  CALL    $RQ1            ; REQUEST BINARY FILE AGAIN
    204 000312  103402                          BCS     30$             ; IF CS, NO ACCESS
    205 000314  112712  000000G                 MOVB    #.EOF,(R2)      ; STORE END OF FILE FUNCTION CODE
    206 000320                          30$:    RETURN
    207                                 ;
    208                                 ; ERROR CONDITIONS
    209                                 ;
    210 000322                          101$:   EMSG$R  22              ; FILE COUNT TOO LARGE
    211 000330                          111$:   EMSG$R  KX              ; RESOURCE ALLOCATION FAILURE
```

```
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  NTLER$

                                ;***
                                ; LOCAL DATA
                                ;***
000000                                  .PSECT  DATA,D

                                        .NLIST  BEX

                                ;
                                ; SYMBOL NAME IN ASCII
                                ;
000000                          ASYM:   .BLKB   6

                                ;
                                ; GLOBAL SYMBOL LOOKUP CONTROL BLOCK
                                ;
000006  000000  000000          GBLK:   .WORD   0,0
000012                          GSYM:   .BLKW   2
000016                          GVAL:   .BLKW   1

                                ;
                                ; GLOBAL SYMBOL LOOKUP ERRORS
                                ;
000020  000032'                 GETAB:  .WORD   $ERR27          ; UNDEFINED SYMBOL (WARNING)
000022  000054'                         .WORD   $ERR29          ; OPEN FAILURE
000024  000112'                         .WORD   $ERR30          ; READ FAILURE
000026  000150'                         .WORD   $ERR31          ; READ OVERFLOW
000030  000176'                         .WORD   $ERR32          ; BAD STB FILE

                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000032                                  NTLER$  ,27,2A,$TWARN,REP2A,$ELINE,<Undefined>
000054                                  NTLER$  ,29,4,$TERR,$PRV.N,$FNOUT,<Open Failure (-***.)>
000112                                  NTLER$  ,30,4,$TERR,$PRV.N,$FNOUT,<Read Failure (-***.)>
000150                                  NTLER$  ,31,4,$TERR,$REP.P,$FNOUT,<Read Overflow>
000176                                  NTLER$  ,32,4,$TERR,$REP.P,$FNOUT,<Format Error>

                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000224  052     040     124     FMT2A:  .ASCIZ  "* Template -- Symbol * "
000254  052     040     123     FMT4:   .ASCIZ  "* Symbol Table File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX
```

```
                                ;+
                                ; $CORE - ".CORE" ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $CVAL=NUMBER OF BLOCKS (MAPPED) OR BYTES (UNMAPPED) NEEDED
                                ;       $VALUE=MAPPING DISPLACEMENT TO BE USED
                                ;       F1.UMR IN $FLAG1 INDICATES WHETHER MEMORY SHOULD BE ALLOCATED
                                ;         FROM UMR OR NON-UMR MEMORY
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0,R1,R2=DESTROYED
                                ;       $NALL INCREMENTED IF ALLOCATION SUCCEEDED
                                ;       F1.ERR SET IF ALLOCATION FAILED
                                ;-
000306                          $CORE::
000306                                  CALL    $VFLV1          ; MAKE SURE IT'S THE PRIMARY TEMPLATE
                                .IF DF  M$$MGE
000312                          CALL    $RQ7            ; REQUEST BINARY FILE
                                .IFF
                                CALL    $RQ5            ; DITTO
                                .ENDC
000316  103451                  BCS     20$             ; IF CS, NO ACCESS
000320  005000                  CLR     R0              ; ASSUME ZERO LENGTH
000322  032767  000000G 000000G BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000330  001402                  BEQ     2$              ; NO ..
000332  005067  000000G         CLR     $CVAL           ; YES .. NO-OP THE ALLOCATION
000336  016701  000000G 2$:     MOV     $CVAL,R1        ; GET BLOCK COUNT
000342  001417                  BEQ     15$             ; IF ZERO, DON'T ALLOCATE
000344                          CALL    MAX14           ; ALLOW ONLY 14. ALLOCATIONS
000350  103434                  BCS     20$             ; IF CS, NO MORE ALLOWED
                                .IF DF  M$$MGE
000352  032767  000000G 000000G BIT     #F1.UMR,$FLAG1  ; ALLOCATE UMR OR NON-UMR MEMORY
000360  001003                  BNE     5$              ; BR IF ALLOCATE UMR SPACE
000362                          CALL    $ALN18          ; ALLOCATE 18 BIT, NON-UMR STORAGE
000366  000402                  BR      10$             ; AND PROCESS
                                .ENDC
000370                  5$:     CALL    $ALL18          ; ALLOCATE 18 BIT, UMR STORAGE
000374  103423          10$:    BCS     ALLERR          ; IF CS, FAILURE
000376  105267  000000G         INCB    $NALL           ; TALLY SUCCESSFUL ALLOCATION
000402  112722  000000C 15$:    MOVB    #.CORE!200,(R2)+ ; STORE FUNCTION CODE
000406  110022                  MOVB    R0,(R2)+        ; STORE FIRST WORD OF ADDRESS
000410  000300                  SWAB    R0              ;       ...
000412  110022                  MOVB    R0,(R2)+        ;       ...
                                .IF DF  M$$MGE
000414  116722  000000G         MOVB    $VALUE,(R2)+    ; STORE SECOND WORD OF ADDRESS
000420  116722  000001G         MOVB    $VALUE+1,(R2)+  ;       ...
                                .ENDC
000424  116722  000000G         MOVB    $CVAL,(R2)+     ; STORE COUNT (FOR CLEANUP ONLY)
000430  116712  000001G         MOVB    $CVAL+1,(R2)    ;       ...
```

```
                                        ;+
                                        ; $ABFI - ACCESS BINARY/LIBRARY FILE ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $FILEN=BINARY FILE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       $BFALL=ALLOCATION SIZE IN 32. WORD BLOCKS
                                        ;       $BFLBN=LOGICAL BLOCK NUMBER
                                        ;       $BFPTR=BINARY BUFFER POINTER
                                        ;       $BFXFR=TRANSFER SIZE IN BYTES
                                        ;       F2.FIL CLEARED IF ANY ERROR OCCURS
                                        ;-
000000                          $ABFI::
000000  032767  000000G 000000G          BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN?
000006  001436                          BEQ     20$             ; IF EQ, NO
                                        .IF DF  M$$MGE
000010  012767  140000  000000G          MOV     #,BASEB,$VALUE  ; SET DEFAULT MAPPING
000016  112767  000006  000000G          MOVB    #DFAPR,$CAPR    ; AND DEFAULT APR NUMBER
                                        .ENDC
000024                                  CALL    $VFLV1          ; MAKE SURE IT'S THE PRIMARY TEMPLATE
                                        .IF DF  M$$MGE
000030                                  CALL    $RQ7            ; REQUEST BINARY BUFFER
                                        .IFF
                                        CALL    $RQ5            ; DITTO
                                        .ENDC
000034  103420                          BCS     15$             ; IF CS, NO ACCESS
000036  010267  000000G                 MOV     R2,$BFPTR       ; SAVE BUFFER POINTER
000042  005067  000000G                 CLR     $BFALL          ; NO FILE SIZE YET
000046  032767  000000G 000000G         BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000054  001013                          BNE     20$             ; YES ..
000056                                  CALL    BN.OPN          ; OPEN BINARY FILE, GET ATTRIBUTES
000062  103405                          BCS     15$             ; IF CS, FAILURE
000064                                  CALL    BN.LBL          ; PROCESS LABEL BLOCK
000070  103005                          BCC     20$             ; IF CC, SUCCESS
000072                          10$:    CALL    $CLDEQ          ; CLOSE BINARY FILE
000076  042767  000000G 000000G 15$:    BIC     #F2.FIL,$FLAG2  ; CLEAR VALID FILE NAME FLAG
000104                          20$:    RETURN
```

LOCAL DATA

```
                                        .SBTTL  LOCAL DATA

                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

                                        .NLIST  BEX


                                ;
                                ; MAPPING DISPLACEMENTS AND APR NUMBERS
                                ;
                                        .IF DF  M$$MGE
000000  000000  020000  040000  MAPP:   .WORD   0,20000,40000,60000,100000,120000,140000,160000
000020     000     001     002  MAPBYT: .BYTE   0,1,2,3,4,5,6,7
                                        .ENDC


                                ;
                                ; DEFAULT APR NUMBER
                                ;
                                        .IF DF  M$$MGE
        000006                  DFAPR=<<.BASEB/2>&77777>/10000
                                        .ENDC


                                ;
                                ; FILE NAME IN ASCII BLANK FILLED TO NINE CHARACTERS
                                ;
000030                          ASCII:  .BLKB   10.
                                ;
                                ; LOCAL COPY OF $FLAGS
                                ;
000042                          .FLAGS: .BLKW   1
```


ERROR MESSAGES

```
                                        .SBTTL  $VXLLC - VERIFY NOT AN LLC

                                ;+
                                ;
                                ; $VXLLC - VERIFY NOT AN LLC
                                ;
                                ; INPUTS:
                                ;       $FLAGS - FLAGS WORD
                                ;
                                ; OUTPUTS:
                                ;       ERROR IF LLC BEING LOADED
                                ;
                                ;-
001176  032767  000000C 000000G $VXLLC: BIT     #FL.DDM!FL.DLC,$FLAGS           ; IS IT DDM OR DLC?
001204  001003                          BNE     10$                             ; BR IF YES
001206                                  EMSG$   90                              ; ELSE ERROR
001214                          10$:    RETURN

        000001                          .END
```

```
                                ;+
                                ; BASE - CHECK ALREADY RESIDENT LIBRARY FOR CORRECT BASE ADDRESS
                                ;
                                ; INPUTS:
                                ;       $LBIAS,$LBADR=DESCRIPTOR ADDRESS
                                ;       $CAPR=USER'S APR NUMBER
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
                                
                                        .IF DF  M$$MGE
                                
000356  016700  000000G         BASE:   MOV     $LBIAS,R0       ; GET DESCRIPTOR BIAS
                                
                                        .IF DF  R$$11M
000362                                  SWSTK$  10$             ;* ENTER KERNEL MODE
                                        .ENDC
                                
000366                                  CALL    $MAPX           ;* MAP TO LIBRARY DESCRIPTOR
000372  016702  000000G                 MOV     $LBADR,R2       ;* GET DESCRIPTOR VIRTUAL ADDRESS
000376  116267  000000G 000000G        MOVB    LB.APR(R2),$BFBAS ;* SAVE BASE APR
                                
                                        .IF DF  R$$11M
000404                                  RETURN                  ;* BACK TO USER MODE
                                        .IFF
                                        CALL    @(SP)+          ;;; RE-ENABLE TASK SWITCHING
                                        .ENDC
                                
000406  126767  000000G 000000G 10$:    CMPB    $BFBAS,$CAPR    ; BUILT FOR CORRECT APR?
000414  001403                         BEQ     20$             ; IF EQ, YES
000416                                  EMSG$   Y2              ; PRINT WARNING MESSAGE
000424                          20$:    RETURN
                                
                                        .ENDC
```

TMPAC8 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES		
BLKSIZ	= 000001		#5-79	6-95	
FMT2	= ******	GX	5-72		
FM.2	= 000000		#5-72		
I$$AS	= ******		6-126		
LBLK	= ******	GX	*6-98		
M$$MGE	= 000000		5-79	5-80	6-100
R$$11D	= ******		6-126		
R$$11M	= 000000		6-104		
S$$BAS	= ******		5-72	5-72	
$ALB	000000	RG	#6-94		
$ALN18	= ******	GX	6-96		
$ELINE	= ******	GX	5-72		
$ERRY1	000000	R	#5-72	6-136	
$LBADR	= ******	GX	*6-120		
$LBERR	= ******	GX	6-135		
$LBIAS	= ******	GX	*6-102		
$MAPX	= ******	GX	6-110		
$REP.C	= * ****	GX	5-72		
$TERR	= ******	GX	5-72		

TMPAC8 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

```
                                        .SBTTL  MACRO DEFINITIONS
                                ;***
                                ; LIBRARY MACROS
                                ;***
                                        .MCALL  EMSG$R,NTLER$,ASR$,EMSG$,SLTDF$
                                        .MCALL  NKRDF$,SAVRG,RESRG

000000                                  SLTDF$                  ; DEFINE SLT OFFSETS AND SYMBOLS
                                        .IF DF  R$$MPL
                                        .MCALL  CTBDF$,KRBDF$

                                        CTBDF$          ; DEFINE CTB OFFSETS
                                        KRBDF$          ; DEFINE KRB OFFSETS
                                        .ENDC

000000                                  NKRDF$          ; DEFINE NETWORK KRB OFFSETS
```

```
                                        .SBTTL  GETIND - GET PDV INDEX
                                ;+
                                ; GETIND - GET PDV INDEX
                                ;
                                ; INPUTS:
                                ;       $PDVA = PDV ADDRESS OF CURRENT PROCESS
                                ;
                                ; OUTPUTS:
                                ;       PDVX = PDV INDEX OF CURRENT PROCESS
                                ;
                                ;-

001136                          GETIND:
001136  010446                         MOV     R4,-(SP)                ; SAVE R4
001140  017746  000000G                 MOV     @PDVNM,-(SP)            ; GET # OF PDV ENTRIES
001144  017704  000000G                 MOV     @PDVTA,R4               ; GET ADDRESS OF PDV MAPPING TABLE

001150  022467  000000G         10$:    CMP     (R4)+,$PDVA             ; IS THIS OUR PDV ?
001154  001403                         BEQ     20$                     ; IF EQ, YES
001156  005316                         DEC     (SP)                    ; ELSE, COUNT DOWN PDV'S
001160  003373                         BGT     10$                     ; IF GT, MORE TO GO
001162                                 CRASH                           ; ELSE, CRASH - CAN'T FIND PDV ENTRY
001164  005726                 20$:    TST     (SP)+                   ; CLEAN UP STACK

001166  167704  000000G                 SUB     @PDVTA,R4               ; CALCULATE PDV INDEX
001172  005744                         TST     -(R4)                   ;
001174  010467  000000'                 MOV     R4,PDVX                 ; SAVE PDV INDEX
001200  012604                         MOV     (SP)+,R4                ; RESTORE R4
001202                                 RETURN
```

TMPALT - TEMPLATE ALLOCATION RO MACRO V05.03b Saturday 29-Jun-85 00:41 Page 18-2
Symbol table

```
ALOCB = ****** GX       ITAB    000004R     002 L$$ASG= 000000          P$$WRD= 000000          $DEAC2= ****** GX
A$$CHK= 000000          I$$RAR= 000000          L$$DRV= 000000          P.REL = ****** GX       $DEA16= ****** GX
A$$CPS= 000000          I$$RDN= 000000          L$$P11= 000001          Q$$OPT= 000010          $ERR33  000012R     002
A$$PRI= 000000          KRBAL   000276R         L$$11R= 000000          RTSPC   000224R         $ERR34  000044R     002
A$$TRP= 000000          K$$CNT= 177546          L.COST  000015          R$$DER= 000000          $ERR35  000112R     002
BLKSIZ= 000001          K$$CSR= 177546          L.CTL   000012          R$$K11= 000001          $ERR36  000150R     002
C$$CKP= 000000          K$$LDC= 000000          L.CVA   177776          R$$SND= 000000          $ERR37  000216R     002
C$$ORE= 000400          K$$TPS= 000074          L.DDM   000002          R$$11M= 000000          $ERR84  000260R     002
C$$RSH= 177564          K.CSR   000002          L.DDS   000004          SF.ACT= 000200          $ERR86  000322R     002
DEACB = ****** GX       K.PRI   000000          L.DLC   000003          SF.ENA= 000100          $ERR9B  000372R     002
D$$BUG= 177514          K.VCT   000001          L.DLM   000006          SF.LPB= 000004          $ERR9C  000432R     002
D$$ISK= 000000          LD$LP = 000000          L.DLS   000010          SF.MFL= 000040          $ERR9Y  000472R     002
D$$L11= 000001          LF.ACT= 100000          L.FLG   000000          SF.PAC= 000020          $FLAGS= ****** GX
D$$YNC= 000000          LF.BRO= 000400          L.KRBA  000016          SF.REA= 000010          $FLAG1= ****** GX
D$$YNM= 000000          LF.BWT= 000007          L.LEN = 000022          SF.SER= 000001          $ICBAD= ****** GX
E$$XPR= 000000          LF.ENA= 002000          L.MPF   000022          SF.SVC= 000002          $ICBLN= ****** GX
FL.CHA= ****** GX       LF.LPB= 001000          L.NMST  000020          SF.UNL= 000040          $ICBSZ= ****** GX
FL.DDM= ****** GX       LF.MDC= 000100          L.NSTA  000014          SLTMA = ****** GX       $KRBLN= ****** GX
FL.LLC= ****** GX       LF.MFL= 004000          L.OWNR  000021          SLTNM = ****** GX       $LDA  = ****** GX
FMT0    000554R     002 LF.MTP= 000020          L.UNT   000013          S$$WRG= 000000          $MAPX = ****** GX
FMT1    000555R     002 LF.PAC= 000200          M$$CRB= 000124          S$$YSZ= 007600          $NALL = ****** GX
FMT3    000560R     002 LF.RDY= 040000          M$$CRX= 000000          S.COST  000001          $NLIB = ****** GX
FM.0  = 000000          LF.REA= 010000          M$$FCS= 000000          S.FLG   000000          $NVECT= ****** GX
FM.1  = 000000          LF.SER= 000040          M$$MGE= 000000          S.LEN   000004          $PCB  = ****** GX
FM.3  = 000000          LF.TIM= 000010          M$$NET= 000000          S.NMST  000002          $PDVA = ****** GX
FM2TMP  000002R     002 LF.UNL= 020000          M$$OVR= 000000          S.OWNR  000003          $PRAVL= ****** GX
F$$LVL= 000001          LF.X2P= 000000          NLBLK   000604R     002 T$$KMG= 000000          $REP.P= ****** GX
F1.INT= ****** GX       LN.CLO= 000000          NTLPT = ****** GX       T$$MIN= 000000          $SIZE = ****** GX
F1.IN1= ****** GX       LN.DUM= 000005          N$$ACC= 000001          UBA     000616R         $SLTA = ****** GX
F1.IN2= ****** GX       LN.LOA= 000004          N$$BUF= 000001          UBLTA   001204R         $TERR = ****** GX
F1.IN3= ****** GX       LN.LOO= 000003          N$$LDV= 000001          V$$CTR= 001000          $TMALT  000000RG
F1.LFI= ****** GX       LN.OAU= 000003          N$$MCP= 000001          XICB    001312R         $TMCHK  000450RG
F1.LFL= ****** GX       LN.OFF= 000001          N$$MLL= 000001          XLT     001272R         $TWE  = ****** GX
F1.MPL= ****** GX       LN.ON = 000000          N$$MOV= 000010          X$$DBT= 000000          $UBIAS= ****** GX
F1.MPT= ****** GX       LN.OOP= 000004          N$$NCT= 000001          ZF.PSE= ****** GX       $UDA  = ****** GX
GETIND  001136R         LN.OPE= 000001          N$$PEM= 000001          Z.AVL = ****** GX       $$CSR = ****** GX
G$$TPP= 000000          LN.REF= 000002          PDVNM = ****** GX       Z.FLG = ****** GX       $$PRI = ****** GX
G$$TSS= 000000          LN.SER= 000002          PDVTA = ****** GX       $ADDR = ****** GX       $$VECT= ****** GX
G$$TTK= 000000          LN.STA= 000017          PDVX    000000R     002 $ALN18= ****** GX       .CXUNL= ****** GX
G$$WRD= 000000          LN.SUB= 000360          P$$P45= 000000          $ALPRO= ****** GX       .DEA16  001372R
HEADR = ****** GX       LN.TRI= 000006

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        001502    001   (RW,I,LCL,REL,CON)
DATA    000606    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 12273 Words  ( 48 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:29.31
SY:TMPALT.V2,[132,134]TMPALT/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETLIB/ML,[130,10]RSXMCM/PA:1,[132,10]TMPALT
```


TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

```
                        .TITLE  IMPEXP - NTL TEMPLATE EXPANSION PROCESSING
                        .IDENT  /V05.00/


;
; COPYRIGHT (C) 1978, 1979, 1980, 1982, 1983, 1985 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;
; MODULE DESCRIPTION:
;
;       NTL - LINE TABLE EXPANSION FRONT END
;
; DISTRIBUTED SYSTEMS SOFTWARE ENGINEERING
;
; IDENT HISTORY:
;
; 1.00  27-FEB-78
;       VERSION 2.0 RELEASE
;
; 2.00  14-DEC-79
;       DECNET-11M/S V3.0
;       DECNET-11M-PLUS V1.0
;
; 3.00  16-APR-82
;       DECNET-11M V3.1
;       DECNET-11M-PLUS V1.1
;
; 4.00  07-NOV-83
;       DECNET-11M V4.0
;       DECNET-11M-PLUS V2.0
;
; 5.00  22-JUL-85
;       DECnet-11M/S V4.2
;       DECnet-11M-Plus V3.0
;       DECnet-Micro/RSX V1.0
;
;
;
;
;
;
```

```
                                        .SBTTL  ADDRB,ADDRW,BLKB,BLKW
                                ;***
                                ; EXPANSION ROUTINES
                                ;***

                                ;
                                ; LOCAL PHYSICAL STATION ADDRESS (BYTE)
                                ;
000046  032767  000000C 000000G ADDRB:  BIT     #FL.DDM!FL.DLC,$FLAGS ; IS IT A DDM OR DLC?
000054  001403                         BEQ     10$             ; BR IF NO
000056  116724  000000G                MOVB    $$LSAD,(R4)+    ; STORE LOCAL STATION ADDRESS
000062                                 RETURN
000064  112724  177777         10$:    MOVB    #-1,(R4)+       ; LSAD NOT VALID FOR LLC
000070                                 RETURN

                                ;
                                ; LOCAL PHYSICAL STATION ADDRESS (WORD)
                                ;
000072  032767  000000C 000000G ADDRW:  BIT     #FL.DDM!FL.DLC,$FLAGS ; IS IT A DDM OR DLC?
000100  001405                         BEQ     10$             ; BR IF NO
000102  005003                         CLR     R3              ; GET LOCAL STATION ADDRESS
000104  156703  000000G                BISB    $$LSAD,R3       ; ...
000110  010324                         MOV     R3,(R4)+        ; STORE IT
000112                                 RETURN
000114  012724  177777         10$:    MOV     #-1,(R4)+       ; LSAD NOT VALID FOR LLC
000120                                 RETURN

                                ;
                                ; ZERO BYTE(S)
                                ;
000122  112500                 BLKB:   MOVB    (R5)+,R0        ; GET BYTE COUNT
000124  105024                 10$:    CLRB    (R4)+           ; STORE A ZERO BYTE
000126  105300                         DECB    R0              ; LOOP BETWEEN 1 AND 256. TIMES
000130  001375                         BNE     10$
000132                                 RETURN

                                ;
                                ; ZERO WORD(S)
                                ;
000134  112500                 BLKW:   MOVB    (R5)+,R0        ; GET WORD COUNT
000136  005024                 10$:    CLR     (R4)+           ; STORE A ZERO WORD
000140  105300                         DECB    R0              ; LOOP BETWEEN 1 AND 256. TIMES
000142  001375                         BNE     10$
000144                                 RETURN
```

```
                                        .SBTTL  X3CH

                                ;
                                ; DEFAULT BLOCK SIZE
                                ;
001672  016724  000000G         X3CH01: MOV     $X3DBS,(R4)+            ; STORE DEFAULT BLOCK SIZE
001676                                  RETURN
                                ;
                                ; MAXIMUM BLOCK SIZE
                                ;
001700  016724  000000G         X3CH02: MOV     $X3MBS,(R4)+            ; STORE MAXIMUM BLOCK SIZE
001704                                  RETURN
                                ;
                                ; DEFAULT WINDOW SIZE
                                ;
001706  116724  000000G         X3CH03: MOVB    $X3DWS,(R4)+            ; STORE DEFAULT WINDOW SIZE
001712                                  RETURN
                                ;
                                ; MAXIMUM WINDOW SIZE
                                ;
001714  116724  000000G         X3CH04: MOVB    $X3MWS,(R4)+            ; STORE MAXIMUM WINDOW SIZE
001720                                  RETURN
                                ;
                                ; CALL TIMER
                                ;
001722  116724  000000G         X3CH05: MOVB    $X3CAL,(R4)+            ; STORE CALL TIMER
001726                                  RETURN
                                ;
                                ; CLEAR TIMER
                                ;
001730  116724  000000G         X3CH06: MOVB    $X3CLR,(R4)+            ; STORE CLEAR TIMER
001734                                  RETURN
                                ;
                                ; RESET TIMER
                                ;
001736  116724  000000G         X3CH07: MOVB    $X3RES,(R4)+            ; STORE RESET TIMER
001742                                  RETURN
                                ;
                                ; RESTART TIMER
                                ;
001744  116724  000000G         X3CH08: MOVB    $X3RST,(R4)+            ; STORE RESTART TIMER
001750                                  RETURN
                                ;
                                ; MAXIMUM CLEAR VALUE
                                ;
001752  116724  000000G         X3CH09: MOVB    $X3MCL,(R4)+            ; STORE MAXIMUM CLEAR VALUE
001756                                  RETURN
                                ;
                                ; MAXIMUM RESETS VALUE
                                ;
001760  116724  000000G         X3CH10: MOVB    $X3MRE,(R4)+            ; STORE MAXIMUM RESETS VALUE
001764                                  RETURN
                                ;
                                ; MAXIMUM RESTARTS VALUE
                                ;
001766  116724  000000G         X3CH11: MOVB    $X3MRS,(R4)+            ; STORE MAXIMUM RESTARTS VALUE
001772                                  RETURN
```

```
    117 000112                          TTAB$   X3CH01,NULL
    118 000114                          TTAB$   X3CH02,NULL
    119 000116                          TTAB$   X3CH03,NULL
    120 000120                          TTAB$   X3CH04,NULL
    121 000122                          TTAB$   X3CH05,NULL
    122 000124                          TTAB$   X3CH06,NULL
    123 000126                          TTAB$   X3CH07,NULL
    124 000130                          TTAB$   X3CH08,NULL
    125 000132                          TTAB$   X3CH09,NULL
    126 000134                          TTAB$   X3CH10,NULL
    127 000136                          TTAB$   X3CH11,NULL
    128 000140                          TTAB$   X2CH01,NULL
    129 000142                          TTAB$   X2CH02,NULL
    130 000144                          TTAB$   X2CH03,NULL
    131 000146                          TTAB$   X2CH04,NULL
    132 000150                          TTAB$   X2CH05,NULL
    133
    134
    135 000000                          .PSECT
```

```
                                        ;+
                                        ; $INT3 - ".INT3" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       NONE
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $VECT3=CURRENT COPY OF $SIZE
                                        ;-
000122                                  $INT3::
000122                                          CALL    $VFDDM          ; MAKE SURE IT'S A DDM
000126  012700  000000G                         MOV     #F1.IN3,R0      ; BIT TO BE SET
000132  012701  000000G                         MOV     #F1.INT,R1      ; BIT WHICH MUST NOT BE SET
000136  012702  000004                          MOV     #4,R2           ; VECTOR INDEX
000142                                          CALLR   VFVCT           ; CHECK VECTOR AVAILABILITY
```

```
                                ;+
                                ; MAX14 - CHECK FOR NO MORE THAN 14. ALLOCATIONS
                                ;
                                ; INPUTS:
                                ;       $NALL = NUMBER OF ALLOCATIONS
                                ;       $NLIB = NUMBER OF LIBRARIES
                                ;
                                ; OUTPUTS:
                                ;       C-BIT = SUCCESS/FAILURE
                                ;       R0 = DESTROYED
                                ;
                                ;-
000336                          MAX14:
000336  116700  000000G                 MOVB    $NALL,R0                ; GET NUMBER OF ALLOCATIONS
000342  010046                          MOV     R0,-(SP)                ; SAVE IT
000344  116700  000000G                 MOVB    $NLIB,R0                ; GET NUMBER OF LIBRARIES
000350  062600                          ADD     (SP)+,R0                ; ADD THE TWO
000352  022700  000015                  CMP     #13.,R0                 ; ARE WE AT THE MAXIMUM ALREADY ?
000356  103401                          BCS     101$                    ; IF CS, NO MORE ALLOWED
000360                                  RETURN
                                ;
                                ; ERROR CONDITION
                                ;
000362                          101$:   EMSG$R  KY                      ; ONLY 14. ALLOCATIONS/LIBRARIES ALLOWED
```

```
                                                                M 3
    113
    114                                 ;
    115                                 ; SYMBOL TABLE UIC AND EXTENSION
    116                                 ;
    117 000304      123     124     102 STBEXT: .ASCIZ  "STB"
        000307      000
    118
    119                                         .EVEN
    120
    121 000000                                  .PSECT
```



```
                                                                N 3
```

```
                                        .IF DF  M$$MGE
000434  062767  000004  000000G          ADD     #4,$SIZE        ; ADJUST DATA BASE SIZE
                                        .IFF
                                        ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
                                        .ENDC

000442                          20$:    RETURN

                                ;
                                ; ERROR CONDITION
                                ;
000444                          ALLERR: EMSG$R  26              ; RESOURCE ALLOCATION FAILURE
```

```
                                        ;+
                                        ; BN.OPN - OPEN BINARY FILE AND GET ATTRIBUTES
                                        ;
                                        ; INPUTS:
                                        ;       $FILEN=BINARY FILE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       C-BIT=SUCCESS/FAILURE
                                        ;       R0,R1,R2=DESTROYED
                                        ;-
000106  005000                  BN.OPN: CLR     R0              ; SETUP ASCIZ FILE SPEC
000110  012701  000000G                 MOV     #$FILEN,R1      ; FOR [131,5X]FILENAME.TSK
000114  005002                          CLR     R2              ;       ...
000116                                  CALL    $AZFS           ;       ...
000122                                  CALL    $CLOPE          ; OPEN BINARY FILE
000126  105777  000000G                TSTB    @$CLIOS         ; SUCCESS? (CLEARS C-BIT)
000132  100403                          BMI     101$            ; IF MI, NO
000134  052010                          BIS     (R0)+,(R0)      ; IS FILE CONTIGUOUS?
000136  001404                          BEQ     111$            ; IF EQ, NO
000140                                  RETURN

                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000142                          101$:   EMSG$R  69              ; OPEN FAILURE
000150                          111$:   CALL    $CLDEQ          ; CLOSE BINARY FILE
000154                                  EMSG$R  71              ; FILE NOT CONTIGUOUS
```

```
                                        .SBTTL  ERROR MESSAGES
                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC
                                
                                        .IF DF  M$$MGE
000044                                  NTLER$  ,39,2,$TWARN,$REP.C,$ELINE,<Invalid Mapping Register>
000106                                  NTLER$  ,ZK,2,$TERR,$REP.C,$ELINE,<UMR's Have Been Disabled>
                                        .ENDC
000150                                  NTLER$  ,89,2,$TERR,$REP.C,$ELINE,<LLC Operator>
000176                                  NTLER$  ,90,2,$TERR,$REP.C,$ELINE,<DDM or DLC Operator>
000232                                  NTLER$  ,91,2,$TERR,$REP.C,$ELINE,<Illegal operand>
                                        .DSABL  LC
                                
                                ;
                                ; ERROR MESSAGE FORMAT STRING
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000262  052     040     124   FMT2:     .ASCIZ  "* Template -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX
                                
000000                                  .PSECT
```

TMPAC6 - NTL TEMPLATE PROCESSIN MACRO V05.03b Saturday 29-Jun-85 00:40 Page 18-1
Symbol table

```
ASCII   000030R     002 G$$TSS= 000000          N$$BUF= 000001          T$$MIN= 000000          $FNR50= ****** GX
A$$CHK= 000000          G$$TTK= 000000          N$$LDV= 000001          UBMPR = 170200          $LBADR= ****** GX
A$$CPS= 000000          G$$WRD= 000000          N$$MCP= 000001          UISAR0= 177640          $LBIAS= ****** GX
A$$PRI= 000000          I$$RAR= 000000          N$$MLL= 000001          UISAR1= 177642          $LIBR1  000106RG
A$$TRP= 000000          I$$RDN= 000000          N$$MOV= 000010          USAR0 = ****** GX       $LINKS  000574RG
BUFUMP= 172354          KISAR0= 172340          N$$NCT= 000001          V$$CTR= 001000          $MAPX = ****** GX
CHECK   000444R         KISAR6= 172354          N$$PEM= 000001          X$$DBT= 000000          $REP.C= ****** GX
CMODE = 140000          KSAR0 = ****** GX       PIRQ  = 177772          X2BHI = 000003          $RQ1  = ****** GX
C$$CKP= 000000          K$$CNT= 177546          PMODE = 030000          X2BLK = 000002          $RQ7  = ****** GX
C$$ORE= 000400          K$$CSR= 177546          PR0   = 000000          X2BLO = 000001          $SIZE = ****** GX
C$$RSH= 177564          K$$LDC= 000000          PR1   = 000040          X2WHI = 000005          $TERR = ****** GX
DFAPR = 000006          K$$TPS= 000074          PR2   = 000100          X2WLO = 000004          $TWARN= ****** GX
D$$BUG= 177514          LB.ADR= ****** GX       PR3   = 000140          X3BHI = 000013          $UMROF  000700RG
D$$ISK= 000000          LB.USE= ****** GX       PR4   = 000200          X3BLO = 000003          $UMRON  000622RG
D$$L11= 000001          LD$LP = 000000          PR5   = 000240          X3WHI = 000002          $VALUE= ****** GX
D$$YNC= 000000          L$$ASG= 000000          PR6   = 000300          X3WLO = 000001          $VFLLC  001156R
D$$YNM= 000000          L$$DRV= 000000          PR7   = 000340          $AZFS = ****** GX       $VFLV1= ****** GX
E$$XPR= 000000          L$$P11= 000001          PS    = 177776          $BFPTR= ****** GX       $VXLLC  001176R
FE.EXT= ****** GX       L$$11R= 000000          P$$P45= 000000          $CAPR = ****** GX       $X2CHB  001034RG
FFN     000430R         MAPBYT  000020R     002 P$$WRD= 000000          $CAT5 = ****** GX       $X2CHW  000710RG
FL.CHA= ****** GX       MAPP    000000R     002 QSTRT = ****** GX       $CORE2  000000RG        $X3CHB  001076RG
FL.DDM= ****** GX       MPAR  = 172100          Q$$OPT= 000010          $ELINE= ****** GX       $X3CHW  000752RG
FL.DLC= ****** GX       MPCSR = 177746          R$$DER= 000000          $ERRZK  000106R     002 .BASEB= 140000
FL.LLC= ****** GX       M$$CRB= 000124          R$$K11= 000001          $ERR39  000044R     002 .CXLBR= ****** GX
FMASK = ****** GX       M$$CRX= 000000          R$$SND= 000000          $ERR89  000150R     002 .CXLB1= ****** GX
FMT2    000262R     002 M$$FCS= 000000          R$$11M= 000000          $ERR90  000176R     002 .FLAGS  000042R     002
FM.2  = 000000          M$$MGE= 000000          SWR   = 177570          $ERR91  000232R     002 .LINKS= ****** GX
F$$LVL= 000001          M$$NET= 000000          S$$WRG= 000000          $FILEN= ****** GX       .PSTCN= ****** GX
F1.UMR= ****** GX       M$$OVR= 000000          S$$YSZ= 007600          $FLAGS= ****** GX       .PSTPT= ****** GX
F2.FIL= ****** GX       NTLPT = ****** GX       TPS   = 177564          $FLAG1= ****** GX       .X2CH1= ****** GX
F2.LBR= ****** GX       N$$ACC= 000001          T$$KMG= 000000          $FLAG2= ****** GX       .X3C01= ****** GX
G$$TPP= 000000

. ABS.  000000    000   (RW,I,GBL,ABS,OVR)
        001216    001   (RW,I,LCL,REL,CON)
DATA    000302    002   (RW,D,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11028 Words  ( 44 Pages)
Size of core pool: 14440 Words  ( 55 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:00:19.24
SY:TMPAC6.V2,[132,134]TMPAC6/CR/-SP=SY:[1,1]RSXMCM.SML/ML,[130,110]NETL1B/ML,[130,10]RSXMCM/PA:1,[132,10]TMPAC6
```

```
                                ;+
                                ; USE - INCREMENT LIBRARY USE COUNT
                                ;
                                ; INPUTS:
                                ;       $LBIAS=DESCRIPTOR BIAS (MAPPED ONLY)
                                ;       $LBADR=DESCRIPTOR VIRTUAL ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       LB.USE, $NLIB INCREMENTED
                                ;-
000426  016702  000000G         USE:    MOV     $BFPTR,R2       ; GET BINARY BUFFER POINTER
000432  112722  000000G                 MOVB    #.LIBR,(R2)+    ; STORE FUNCTION CODE
                                
                                        .IF DF  M$$MGE
000436  116722  000000G                 MOVB    $LBIAS,(R2)+    ; STORE DESCRIPTOR BIAS
000442  116722  000001G                 MOVB    $LBIAS+1,(R2)+  ;       ...
                                        .ENDC
                                
000446  116722  000000G                 MOVB    $LBADR,(R2)+    ; STORE DESCRIPTOR ADDRESS
000452  116722  000001G                 MOVB    $LBADR+1,(R2)+  ;       ...
                                
                                        .IF DF  M$$MGE
000456  116722  000000G                 MOVB    $VALUE,(R2)+    ; STORE USER'S MAPPING TO LIBRARY
000462  116712  000001G                 MOVB    $VALUE+1,(R2)   ;       ...
000466  062767  000004  000000G         ADD     #4,$SIZE        ; ADJUST LINE TABLE SIZE
000474  032767  000000G 000000G         BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000502  001015                         BNE     20$             ; YES .. DON'T UPDATE USE COUNT
000504  016700  000000G                 MOV     $LBIAS,R0       ; GET DESCRIPTOR BIAS
                                        .IF DF  R$$11M
000510                                  SWSTK$  10$             ;* ENTER KERNEL MODE
                                        .ENDC
000514                                  CALL    $MAPX           ;* MAP TO THE DESCRIPTOR
000520  016702  000000G                 MOV     $LBADR,R2       ;* GET DESCRIPTOR ADDRESS
000524  105262  000000G                 INCB    LB.USE(R2)      ;* UPDATE USE COUNT (NOTE - BYTE FIELD)
                                        .IF DF  R$$11M
000530                                  RETURN                  ;* BACK TO USER MODE
                                        .IFF
                                        CALL    @(SP)+          ;* CALL CO-ROUTINE TO RESTORE MAPPING
                                        .ENDC
                                .IFF
                                CLRB    (R2)+           ; DEPOSIT ZERO BYTES
                                CLRB    (R2)            ;       ...
                                ADD     #2,$SIZE        ; ADJUST LINE TABLE SIZE
                                BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
                                BNE     20$             ; YES .. DON'T UPDATE USE COUNT
                                MOV     $LBADR,R2       ; GET DESCRIPTOR ADDRESS
                                INC     LB.USE(R2)      ; UPDATE USE COUNT
                                .ENDC
                                
000532  105267  000000G         10$:    INCB    $NLIB           ; TALLY ALL ".LIBR"
000536                          20$:    RETURN
```

TMPAC8 CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

MACRO CROSS REFERENCE CREF 04.00

MACRO NAME	REFERENCES			
CALL	6-96	6-106	6-110	6-135
EMSG$R	#5-59	6-136		
NTLER$	#5-59	5-72		
RETURN	6-130			
SWSTK$	6-106			

FILEID**TMPAC9

```
                                        .SBTTL  LOCAL DATA
                                ;***
                                ; LOCAL DATA
                                ;***

000000                                  .PSECT  DATA,D

000000                          PDVX:   .BLKW   1               ; PDV INDEX OF PROCESS
000002                          FM2TMP: .BLKW   1               ; $FMASK+2 TEMP LOCATION
                                        .IF DF R$$MPL
                                ;
                                ; ICB ALLOCATION VARIABLES
                                ;
                                KRBO:   .BLKW   1               ; BIT SET FOR EACH CPU AN ICB WAS ALLOCATED FOR
                                CPU:    .BLKW   1               ; CPU NUMBER OF CURRENTLY ALLOCATED ICB
                                BIAS:   .BLKW   1               ; POOL BIAS OF LAST ICB ALLOCATED
                                K6TMP:  .BLKW   1               ; TEMPORARY $K6TAB LOCATION
                                        .ENDC   ; DF R$$MPL


                                ;
                                ; INTERRUPT BITS TABLE
                                ;
000004  000000G                 ITAB:   .WORD   F1.IN1
000006  000000C                         .WORD   F1.IN1!F1.IN2
000010  000000C                         .WORD   F1.IN1!F1.IN2!F1.IN3
```

```
                                        .SBTTL  UBLTA - STORE LINE TABLE ADDRESS
                                ;
                                ; STORE LINE TABLE ADDRESS DATA IN UNLOAD BLOCK
                                ;
001204                          UBLTA:
001204  011167  000000G                 MOV     (R1),$UBIAS     ;* SET UNLOAD BLOCK BIAS
001210  116723  000000'                 MOVB    PDVX,(R3)+      ;* STORE PDV INDEX OF THIS PROCESS
001214  105023                          CLRB    (R3)+           ;* CLEAR RESERVED BYTE
001216  016723  000000G                 MOV     $ADDR,(R3)+     ;* STORE LINE TABLE ADDRESS
001222  016713  000000G                 MOV     $SIZE,(R3)      ;* AND LINE TABLE SIZE
                                
001226  005203                          INC     R3              ;* MAKE IT AN ODD ADDRESS (TEMPORARY)
001230  005767  000000G                 TST     $NALL           ;* ANY ALLOCS/LIBRS?
001234  001404                          BEQ     30$             ;* IF ZERO, NO
001236  105243                          INCB    -(R3)           ;* INDICATE SOME ALLOCS/LIBRS
001240  005723                          TST     (R3)+           ;* SKIP LINE TABLE SIZE
001242  016723  000000G                 MOV     $NALL,(R3)+     ;* STORE ALLOC/LIBR COUNTS
                                
001246  010367  000000G          30$:   MOV     R3,$UDA         ;* SET UNLOAD DATA VIRTUAL ADDRESS
001252  116704  000000G                 MOVB    $NALL,R4        ;* GET NUMBER OF ALLOCS
001256  006304                          ASL     R4              ;* MULTIPLY BY 4 (2 WORDS PER ENTRY)
001260  006304                          ASL     R4              ;*
001262  060304                          ADD     R3,R4           ;* ADD UNLOAD BLOCK ADDRESS
001264  010467  000000G                 MOV     R4,$LDA         ;* GIVING LIBRARY DATA ADDRESS
001270                                  RETURN
```

TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 1

SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES					
ALOCB	= ******	GX	8-255	9-331				
BLKSIZ	= 000001		#7-145	11-580				
DEACB	= ******	GX	14-778					
FL.CHA	= ******	GX	8-170					
FL.DDM	= ******	GX	8-172	8-209	10-446	14-713		
FL.LLC	= ******	GX	8-297					
FMT0	000554	R	7-112	#7-121				
FMT1	000555	R	7-103	7-104	7-109	7-110	7-111	#7-122
FMT3	000560	R	7-105	7-106	7-107	7-108	#7-123	
FM.0	= 000000		#7-112					
FM.1	= 000000		#7-103	#7-104	#7-109	#7-110	#7-111	
FM.3	= 000000		#7-105	#7-106	#7-107	#7-108		
FM2TMP	000002	R	#6-79					
F1.INT	= ******	GX	10-453					
F1.IN1	= ******	GX	6-93	6-94	6-95	10-457		
F1.IN2	= ******	GX	6-94	6-95	10-457			
F1.IN3	= ******	GX	6-95	10-457				
F1.LFI	= ******	GY	10-466					
F1.LFL	= ******	GX	10-464					
F1.MPL	= ******	GX	10-460					
F1.MPT	= ******	GX	10-462					
GETIND	001136	R	11-497	#12-652				
HEADR	= ******	GX	8-272					
ITAB	000004	R	#6-93	10-458				
KRBAL	000276	R	8-177	#9-326				
K.CSR	000002		*9-367					
K.PRI	000000		*9-362					
K.VCT	000001		*9-366					
LF.ACT	= 100000		#5-61					
LF.BRO	= 000400		#5-61					
LF.BWT	= 000007		#5-61					
LF.ENA	= 002000		#5-61					
LF.LPB	= 001000		#5-61					
LF.MDC	= 000100		#5-61					
LF.MFL	= 004000		#5-61					
LF.MTP	= 000020		#5-61					
LF.PAC	= 000200		#5-61					
LF.RDY	040000		#5-61					
LF.REA	= 010000		#5-61					
LF.SER	= 000040		#5-61					
LF.TIM	= 000010		#5-61					
LF.UNL	= 020000		#5-61					
LF.X2P	= 000000		#5-61					
LN.CLO	= 000000		#5-61					
LN.DUM	= 000005		#5-61					
LN.LOA	= 000004		#5-61					
LN.LOO	= 000003		#5-61					
LN.OAU	= 000003		#5-61					
LN.OFF	= 000001		#5-61					
LN.ON	= 000000		#5-61					
LN.OOP	= 000004		#5-61					
LN.OPE	= 000001		#5-61					

TMPALT CREATED BY MACRO ON 29-JUN-85 AT 00:41 PAGE 2

```
                                        ;+
                                        ; $TMEXP - FRONT END FOR BINARY TO REAL PROCESSING
                                        ;
                                        ; THIS ROUTINE CALLS $TMEX2 IF NO ERRORS HAVE OCCURRED DURING TEMPLATE
                                        ; PROCESSING AND $TMEX3 OTHERWISE.
                                        ;
                                        ; INPUTS:
                                        ;       $ADDR=ADDRESS OF WHERE REAL DATA GOES
                                        ;       $NALL=NUMBER OF ALLOCATIONS AND LIBRARIES THIS TEMPLATE
                                        ;       $FLAGS INDICATES WHETHER PROCESS IS DLC ONLY OR NOT (MAPPED ONLY)
                                        ;       F1.ERR SET IF ANY ERRORS OCCURRED
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL REGISTERS=DESTROYED
                                        ;-
 71 000000                              $TMEXP::
 72 000000  022767  177777  000000G     10$:    CMP     #-1,$ADDR               ;IS IT EMPTY LLC DATA BASE?
 73 000006  001440                              BEQ     30$                     ;BR IF YES
 74 000010  005767  000000G                     TST     $FLAG1                  ;DID AN ERROR OCCUR?
 75 000014  100416                              BMI     20$                     ;BR IF YES
 76 000016  032767  [illegible]00G 000000G     BIT     #FL.CHA,$FLAGS          ; MUX CHARACTERISTICS UPDATE ONLY ?
 77 000024  001020                              BNE     25$                     ; YES ..
 78
 79                                             .IF DF  M$$MGE
 80
 81                                             .IF DF  R$$11M
 82 000026  026727  000000G 120000              CMP     $ADDR,#120000           ;LINE TABLE AT END OF PROCESS PARTITION?
 83 000034  103403                              BLO     10$                     ;IF LO, NO
 84 000036                                      CALL    $TMEX6                  ;EXPAND LINE TABLE AT END OF PARTITION
 85 000042  000422                              BR      30$                     ;AND LEAVE
 86
 87                                             .IFF
 88
 89                                             CMP     $ADDR,#60000            ;LINE TABLE AT END OF PROCESS PARTITION?
 90                                             BHIS    10$                     ;BR IF NO
 91                                             CALL    $TMEX6                  ;EXPAND LINE TABLE AT END OF PARTITION
 92                                             BR      30$                     ;AND LEAVE
 93                                             .ENDC
 94
 95                                             .IFTF
 96
 97 000044                              10$:    CALL    $TMEX2                  ;EXPAND THE LINE TABLE
 98 000050  000417                              BR      30$                     ;AND LEAVE
 99 000052  005767  000000G             20$:    TST     $NALL                   ;ANYTHING TO CLEAN UP?
100 000056  001414                              BEQ     30$                     ;BR IF NO
101 000060                                      CALL    $TMEX3                  ;DO THE CLEANUP
102 000064  000411                              BR      30$
103
104 000066                              25$:
105                                             .IFT
106
107                                             .IF DF  R$$11M
108 000066  026727  000000G 120000              CMP     $ADDR,#120000           ;LINE TABLE AT END OF PROCESS PARTITION?
109 000074  103403                              BLO     27$                     ;IF LO, NO
110 000076                                      CALL    $TMEX7                  ; UPDATE THE LINE CHARACTERISTICS
111 000102  000402                              BR      30$                     ;AND LEAVE
112
```

```
                                ;
                                ; SINGLE BYTE
                                ;
000146  112524                 BYTE:   MOVB    (R5)+,(R4)+     ; COPY A BYTE
000150                                  RETURN
                                ;
                                ; CSR
                                ;
000152  112500                  CSR:    MOVB    (R5)+,R0        ; GET MODIFICATION VALUE
000154  066700  000000G                 ADD     $$CSR,R0        ; ADD IN CSR ADDRESS
000160  010024                          MOV     R0,(R4)+        ; STORE THE RESULT
000162                                  RETURN
                                ;
                                ; COUNTER TIMER
                                ;
000164  016724  000000G         CTIM:   MOV     $CTIM,(R4)+     ; STORE THE COUNTER TIMER VALUE
000170                                  RETURN
                                ;
                                ; DEVICE CHARACTERISTICS
                                ;
000172  005000                  DVCHA:  CLR     R0              ; ASSUME DLC
000174  032767  000000G 000000G         BIT     #FL.DDM,$FLAGS  ; IS IT A DDM?
000202  001414                          BEQ     10$             ; IF EQ, NO
000204  032767  000000G 000000G         BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
000212  001410                          BEQ     10$             ; IF EQ, NO
000214  005767  000154'                 TST     FLAGS           ; IS IT MUX EXPANSION?
000220  100020                          BPL     101$            ; IF PL, NO
000222  016700  000152'                 MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
000226  006300                          ASL     R0              ; CONVERT TO A DOUBLEWORD INDEX
000230  006300                          ASL     R0              ;       ...
000232  022020                          CMP     (R0)+,(R0)+     ; SKIP DLC CHARACTERISTICS
000234                          10$:    SAVRG   R3              ; SAVE REG
000236  016703  000000G                 MOV     $SLTSV,R3       ; COPY SLT ADDRESS
000242  010463  177776                  MOV     R4,L.CVA(R3)    ; COPY FIRST DEVICE CHAR. WORD VIRTUAL ADDR.
000246                                  RESRG   R3              ; RESTORE REG
000250  016024  000000G                 MOV     $$DCHA(R0),(R4)+ ; COPY DEVICE CHARACTERISTICS
000254  016024  000002G                 MOV     $$DCHA+2(R0),(R4)+ ; TWO WORDS
000260                                  RETURN
000262  012724  177777          101$:   MOV     #-1,(R4)+       ; DDM DVCHA NOT IN MXTAB FILE
000266  012724  177777                  MOV     #-1,(R4)+       ;       ...
000272                                  RETURN
                                ;
                                ; PROTOCOL EMULATOR CHARACTERISTICS
                                ;
000274  005000                  PECHA:  CLR     R0              ; ASSUME DLC
000276  032767  000000G 000000G         BIT     #FL.DDM,$FLAGS  ; IS IT A DDM?
000304  001413                          BEQ     10$             ; IF EQ, NO
000306  032767  000000G 000000G         BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
000314  001407                          BEQ     10$             ; IF EQ, NO
000316  005767  000154'                 TST     FLAGS           ; IS IT MUX EXPANSION?
000322  100004                          BPL     10$             ; IF PL, NO
000324  016700  000152'                 MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
000330  006300                          ASL     R0              ; CONVERT TO A WORD INDEX
000332  005720                          TST     (R0)+           ; SKIP DLC CHARACTERISTICS
```

```
                                        .SBTTL  X2CH
                                ;
                                ; MAXIMUM WINDOW SIZE
                                ;
002000'                         X2CH01: CLR     R0              ; ASSUME NOT A MUX EXPANSION
001774  005000
001776  032767  000000G 000000G        BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
002004  001403                        BEQ     10$             ; IF EQ, NO
002006  016700  000152'               MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
002012  006300                        ASL     R0              ; CONVERT TO A WORD INDEX
002014  116024  000000G         10$:  MOVB    $X2MWS(R0),(R4)+ ; STORE MAXIMUM WINDOW SIZE
002020                                RETURN
                                ;
                                ; MAXIMUM BLOCK SIZE
                                ;
002022  005000                  X2CH02: CLR     R0              ; ASSUME NOT A MUX EXPANSION
002024  032767  000000G 000000G        BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
002032  001403                        BEQ     10$             ; IF EQ, NO
002034  016700  000152'               MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
002040  006300                        ASL     R0              ; CONVERT TO A WORD INDEX
002042  016024  000000G         10$:  MOV     $X2MBS(R0),(R4)+ ; STORE MAXIMUM BLOCK SIZE
002046                                RETURN
                                ;
                                ; MAXIMUM RETRANSMIT COUNT
                                ;
002050  005000                  X2CH03: CLR     R0              ; ASSUME NOT A MUX EXPANSION
002052  032767  000000G 000000G        BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
002060  001403                        BEQ     10$             ; IF EQ, NO
002062  016700  000152'               MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
002066  006300                        ASL     R0              ; CONVERT TO A WORD INDEX
002070  116024  000000G         10$:  MOVB    $X2MRT(R0),(R4)+ ; STORE MAXIMUM RETRANSMIT COUNT
002074                                RETURN
                                ;
                                ; RETRANSMIT TIMER
                                ;
002076  005000                  X2CH04: CLR     R0              ; ASSUME NOT A MUX EXPANSION
002100  032767  000000G 000000G        BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
002106  001403                        BEQ     10$             ; IF EQ, NO
002110  016700  000152'               MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
002114  006300                        ASL     R0              ; CONVERT TO A WORD INDEX
002116  016024  000000G         10$:  MOV     $X2RET(R0),(R4)+ ; STORE RETRANSMIT TIMER
002122                                RETURN
                                ;
                                ; HOLDBACK TIMER
                                ;
002124  005000                  X2CH05: CLR     R0              ; ASSUME NOT A MUX EXPANSION
002126  032767  000000G 000000G        BIT     #FL.MUX,$FLAGS  ; IS IT A MUX DEVICE?
002134  001403                        BEQ     10$             ; IF EQ, NO
002136  016700  000152'               MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
002142  006300                        ASL     R0              ; CONVERT TO A WORD INDEX
002144  016024  000000G         10$:  MOV     $X2HLD(R0),(R4)+ ; STORE HOLDBACK TIMER
002150                                RETURN

        000001                          .END
```

```
                                        ;+
                                        ; $TMEX3 - CLEANUP BINARY INFORMATION AFTER TEMPLATE ERROR
                                        ;
                                        ; THIS ROUTINE IS CALLED ONLY IF A TEMPLATE ERROR OCCURS. IT DEALLOCATES
                                        ; ALLOCATIONS AND LIBRARY REFERENCES MADE EARLIER IN THE TEMPLATE.
                                        ;
                                        ; INPUTS:
                                        ;       $BIN=START OF BINARY INFORMATION
                                        ;       $PTR=END OF BINARY INFORMATION
                                        ;
                                        ; OUTPUTS:
                                        ;       ALL REGISTERS=DESTROYED
                                        ;-
000000                                  $TMEX3::
000000  016705  000000G                        MOV     $BIN,R5         ; START OF BINARY DATA
000004  026705  000000G                10$:    CMP     $PTR,R5         ; IS IT END OF BINARY DATA?
000010  001407                                BEQ     20$             ; IF EQ, YES
000012  005004                                CLR     R4              ; GET FUNCTION CODE
000014  152504                                BISB    (R5)+,R4        ;  ...
000016  001404                                BEQ     20$             ; IF ZERO, END OF FILE
000020  106304                                ASLB    R4              ; CONVERT TO A WORD INDEX
000022                                         CALL    @TABLE-2(R4)    ; EXPAND ONE FIELD
000026  000766                                BR      10$             ; KEEP LOOPING UNTIL END
000030                                 20$:    RETURN
```

```
                                        ;+
                                        ; VFVCT - CHECK VECTOR AVAILABILITY
                                        ;
                                        ; INPUTS:
                                        ;       R0=BIT TO BE SET IN $FLAG1
                                        ;       R1=BITS WHICH MUST NOT BE SET IN $FLAG1
                                        ;       R2=DISPLACEMENT FROM FIRST VECTOR
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       F1.ERR SET IF VECTOR NOT AVAILABLE
                                        ;       $NVECT INCREMENTED
                                        ;-
000146                                  VFVCT:  CALL    $VF  C          ; MAKE SURE THIS IS THE ONLY OCCURANCE
000152  103436                                 BCS     10$             ; IF CS, IT ISN'T
000154  030167  000000G                        BIT     R1,$FLAG1       ; ARE ANY ILLEGAL BITS SET?
000160  001034                                 BNE     111$            ; IF NE, YES
000162  016762  000000G 000000G                MOV     $SIZE,$VECT1(R2) ; STORE CURRENT DISPLACEMENT
000170  006302                                 ASL     R2              ; CONVERT TO A DOUBLEWORD INDEX
000172  066702  000000G                        ADD     $$VECT,R2       ; GIVING REAL VECTOR ADDRESS
000176  026702  000000G                        CMP     $MXVCT,R2       ; LEGAL VECTOR?
000202  101426                                 BLOS    121$            ; IF LOS, NO
000204  032767  000000G 000000G                BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000212  001014                                 BNE     5$              ; YES ..

000214  010200                                 MOV     R2,R0           ; COPY VECTOR ADDRESS
000216  042700  000077                        BIC     #77,R0          ; MAKE IT A MULTIPLE OF 100
000222                                         ASR$    5,R0            ; CONVERT TO A WORD INDEX
                                                                       ; BY SHIFTING RIGHT 5 PLACES


                                        .IF DF  R$$MPL

                                        TSTB    @NCPU           ; MULTIPROCESSOR SYSTEM?
                                        BEQ     3$              ; BR IF NO
                                        MOV     $URM,SETAF+S.AFAF ; GET UNIBUS RUN MASK
                                        DIR$    #SETAF          ; SET AFFINITY
                                        BCS     131$            ; BR IF FAILED
                                        TST     $DSW            ; SUCCESSFUL?
                                        BMI     131$            ; BR IF NO
                                3$:     CMP     NONSI,(R2)      ; IS VECTOR IN USE ?
                                        BEQ     4$              ; IF NOT - BRANCH

                                        .ENDC
000234  016001  000000'                 MOV     NSITAB(R0),R1   ; GET ERROR LOG VECTOR ADDRESS
000240  021112                          CMP     (R1),(R2)       ; IS VECTOR IN USE?
000242  001011                          BNE     131$            ; IF NE, YES
000244                                  4$:

                                           .IF DF R$$MPL

                                        TSTB    @NCPU           ; MULTIPROCESSOR SYSTEM?
                                        BEQ     5$              ; BR IF NO
                                        RMAF$S                  ; REMOVE AFFINITY

                                           .ENDC

000244  005267  000000G                5$:     INC     $NVECT          ; TALLY NUMBER OF TIMES CALLED
000250                                  10$:    RETURN
```

```
                                        ;+
                                        ; $MPTB - ".MPTAB" ACTION ROUTINE
                                        ; $LFILE - ".LFILE" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $$MTP/$VALUE=REPEAT COUNT
                                        ;       $FILEN=SECONDARY TEMPLATE NAME
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;-

                                                .ENABL  LSB

000370                                  $MPTB::
000370                                          CALL    $VFDLC          ; MAKE SURE IT'S A DLC
000374  012700  000000G                         MOV     #F1.MPT,R0      ; MAKE SURE THIS IS THE ONLY ONE
000400                                          CALL    $VFSPC          ;       ...
000404  103500                                  BCS     40$             ; IF CS, ERROR
000406  032767  000000G 000000G                BIT     #F1.MPL,$FLAG1  ; HAS ".MPLHD" BEEN SEEN?
000414  001475                                  BEQ     101$            ; IF EQ, NO
000416  012700  000000G                         MOV     #.MPTAB,R0      ; SET FUNCTION CODE
000422  016701  000000G                         MOV     $MPLD,R1        ; GET ".MPLHD" DISPLACEMENT
000426  016767  000000G 000000G                MOV     $$MTP,$VALUE    ; DO IT <STATION COUNT> TIMES
000434  000415                                  BR      10$
000436                                  $LFILE::
000436  012700  000000G                         MOV     #F1.LFI,R0      ; MAKE SURE THIS IS THE ONLY ONE
000442                                          CALL    $VFSPC          ;       ...
000446  103457                                  BCS     40$             ; IF CS, ERROR
000450  032767  000000G 000000G                BIT     #F1.LFL,$FLAG1  ; HAS ".LFLHD" BEEN SEEN?
000456  001457                                  BEQ     111$            ; IF EQ, YES
000460  012700  000000G                         MOV     #.LFILE,R0      ; SET FUNCTION CODE
000464  016701  000000G                         MOV     $LFLD,R1        ; GET ".LFLHD" DISPLACEMENT
000470  022767  000400  000000G 10$:    CMP     #256.,$VALUE    ; IS COUNT > 256.?
000476  103452                                  BLO     CTLERR          ; IF LO, YES
000500  005767  000000G                         TST     $VALUE          ; IS COUNT 0?
000504  001440                                  BEQ     40$             ; IF EQ, YES
000506                                          CALL    $RQ4            ; REQUEST BINARY FILE
000512  103435                                  BCS     40$             ; IF CS, NO ACCESS
000514  110022                                  MOVB    R0,(R2)+        ; STORE FUNCTION CODE
000516  116722  000000G                         MOVB    $VALUE,(R2)+    ; FILE REPEAT COUNT
000522  110122                                  MOVB    R1,(R2)+        ; AND LISTHEAD DISPLACEMENT
000524  000301                                  SWAB    R1              ;       ...
000526  110112                                  MOVB    R1,(R2)         ;       ...
000530  016746  000000G                         MOV     $SIZE,-(SP)     ; SAVE CURRENT LINE TABLE SIZE
000534                                          CALL    $FILE           ; PROCESS SECONDARY TEMPLATE
000540                                          CALL    $EVEN           ; AUTOMATIC WORD RE-ALIGNMENT
000544  012600                                  MOV     (SP)+,R0        ; RESTORE SIZE FROM START OF FILE
000546                                          CALL    $RQ1            ; REQUEST BINARY FILE AGAIN
000552  103415                                  BCS     40$             ; IF CS, NO ACCESS
000554  112712  000000G                         MOVB    #.EOF,(R2)      ; STORE END OF FILE FUNCTION CODE
000560  166700  000000G                         SUB     $SIZE,R0        ; GIVING AMOUNT ADDED BY SECONDARY TEMPLATE
000564  001005                                  BNE     30$             ; IF NE, OK
000566                                          EMSG$R  43              ; SECONDARY TEMPLATE EMPTY
000574  160067  000000G          20$:    SUB     R0,$SIZE        ; ADJUST DATA BASE SIZE
000600  005367  000000G          30$:    DEC     $VALUE          ; BY <FILE COUNT-1> * <FILE SIZE>
000604  001373                                  BNE     20$             ;       ...
```

```
                                        ;+
                                        ; $GLOBL - GLOBAL SYMBOL ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       .PSTPT=SYMBOL ADDRESS
                                        ;       .PSTCN=SYMBOL LENGTH
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $VALUE=EXTERNAL SYMBOL VALUE
                                        ;-
000000                                  $GLOBL::
000000  005067  000000G                         CLR     $VALUE          ; SET NO VALUE YET
000004  032767  000000G 000000G                 BIT     #F1.STE,$FLAG1  ; ANY SYMBOL TABLE ERRORS YET?
000012  001063                                  BNE     50$             ; IF NE, YES
000014  016700  000000G                         MOV     .PSTPT,R0       ; GET SYMBOL ADDRESS
000020  012701  000000'                         MOV     #ASYM,R1        ; GET LOCAL SCRATCH PAD ADDRESS
000024  016702  000000G                         MOV     .PSTCN,R2       ; GET SYMBOL LENGTH
000030  022702  000006                         CMP     #6,R2           ; 6 CHARACTERS OR LESS?
000034  103002                                  BHIS    10$             ; IF HIS, YES
000036  012702  000006                         MOV     #6,R2           ; USE ONLY FIRST 6 CHARACTERS OF SYMBOL
000042  010246                         10$:    MOV     R2,-(SP)        ; SAVE LENGTH
000044  112021                         20$:    MOVB    (R0)+,(R1)+     ; COPY SYMBOL
000046                                          SOB     R2,20$
000052  012702  000006                         MOV     #6,R2           ; IS SYMBOL LENGTH 6?
000056  162602                                  SUB     (SP)+,R2
000060  001404                                  BEQ     40$             ; IF EQ, YES
000062  112721  000040                 30$:    MOVB    #' ,(R1)+       ; FILL WITH BLANKS
000066                                          SOB     R2,30$          ; UP TO A LENGTH OF 6
000072  012700  000000'                 40$:    MOV     #ASYM,R0        ; CONVERT 6 CHARACTERS
000076  010701                                  MOV     PC,R1           ; FROM ASCII TO RAD50
000100                                          CALL    $CAT5           ; AND STORE IN CONTROL BLOCK
000104  010167  000012'                         MOV     R1,GSYM         ;       ...
000110  010701                                  MOV     PC,R1           ;       ...
000112                                          CALL    $CAT5           ;       ...
000116  010167  000014'                         MOV     R1,GSYM+2       ;       ...
000122  005000                                  CLR     R0              ; SETUP ASCIZ FILE SPEC
000124  005001                                  CLR     R1              ; NOTE - [1,1]XXX.STB FOR 11D/IAS
000126  012702  000304'                         MOV     #STBEXT,R2      ;       ...
000132                                          CALL    $AZFS           ;       ...
000136  012700  000006'                         MOV     #GBLK,R0        ; POINT TO SYMBOL CONTROL BLOCK
000142                                          CALL    $SYMBL          ; LOOK FOR GLOBAL SYMBOL
000146  103406                                  BCS     101$            ; IF CS, FILE ERROR OF SOME SORT
000150  016767  000016' 000000G                 MOV     GVAL,$VALUE     ; SET SYMBOL VALUE
000156  005701                                  TST     R1              ; CHECK COUNT OF UNDEFINEDS
000160  001010                                  BNE     111$            ; IF NE, SYMBOL WAS UNDEFINED
000162                                  50$:    RETURN
                                        ;
                                        ; ERROR CONDITIONS
                                        ;
000164  126727  000000G 000006         101$:    CMPB    $CLEVL,#6       ; IS GCL AT MAXIMUM DEPTH?
000172  001403                                  BEQ     111$            ; IF EQ, YES
000174  052767  000000G 000000G                 BIS     #F1.STE,$FLAG1  ; INDICATE SYMBOL TABLE ERROR
000202  016000  000020'                 111$:   MOV     GETAB-2(R0),R0  ; GET ERROR MESSAGE BLOCK ADDRESS
000206                                          CALL    @(R0)+          ; PRINT ERROR MESSAGE
```

```
                                        ;+
                                        ; $SCOM - ".POOL" OR ".SCOM" ACTION ROUTINE
                                        ;
                                        ; INPUTS:
                                        ;       $VALUE=NUMBER OF BYTES NEEDED
                                        ;
                                        ; OUTPUTS:
                                        ;       R0,R1,R2=DESTROYED
                                        ;       $NALL INCREMENTED IF ALLOCATION SUCCEEDED
                                        ;       F1.ERR SET IF ALLOCATION FAILED
                                        ;-
000452                                  $SCOM::
000452                                          CALL    $VFLV1          ; MAKE SURE IT'S THE PRIMARY TEMPLATE
000456                                          CALL    $RQ5            ; REQUEST BINARY FILE
000462  103443                                 BCS     20$             ; IF CS, NO ACCESS
000464  005000                                 CLR     R0              ; ASSUME ZERO LENGTH
000466  032767  000000G 000000G                BIT     #FL.CHA,$FLAGS  ; MUX CHARACTERISTICS UPDATE ONLY ?
000474  001402                                 BEQ     5$              ; NO ..
000476  005067  000000G                        CLR     $VALUE          ; YES .. NO-OP THE ALLOCATION
000502  016701  000000G                5$:     MOV     $VALUE,R1       ; GET BYTE COUNT
000506  001415                                 BEQ     10$             ; IF ZERO, DON'T ALLOCATE
000510                                          CALL    MAX14           ; ALLOW ONLY 14. ALLOCATIONS
000514  103426                                 BCS     20$             ; IF CS, NO MORE ALLOWED
000516  005201                                 INC     R1              ; ROUND COUNT UP TO MULTIPLE OF 2
000520  042701  000001                        BIC     #1,R1           ;       ...
000524  010167  000000G                        MOV     R1,$VALUE       ; RESET COUNT VALUE
000530                                          CALL    $ALL16          ; ALLOCATE 16 BIT ADDRESSABLE STORAGE
000534  103743                                 BCS     ALLERR          ; IF CS, FAILURE
000536  105267  000000G                        INCB    $NALL           ; TALLY SUCCESSFUL ALLOCATION
000542  112722  000000G                10$:     MOVB    #.SCOM,(R2)+    ; STORE FUNCTION CODE
000546  110022                                 MOVB    R0,(R2)+        ; STORE ADDRESS
000550  000300                                 SWAB    R0
000552  110022                                 MOVB    R0,(R2)+
000554  116722  000000G                        MOVB    $VALUE,(R2)+    ; STORE COUNT (FOR CLEANUP ONLY)
000560  116712  000001G                        MOVB    $VALUE+1,(R2)
000564  062767  000002  000000G               ADD     #2,$SIZE        ; ADJUST DATA BASE SIZE
000572                                  20$:    RETURN
```

```
                                ;+
                                ; BN.LBL - READ AND PROCESS LABEL BLOCK
                                ;
                                ; INPUTS:
                                ;       NONE
                                ;
                                ; OUTPUTS:
                                ;       C-BIT=SUCCESS/FAILURE
                                ;       R0,R1,R2=DESTROYED
                                ;       $BFALL=SIZE OF ALLOCATION NEEDED
                                ;       $BFLBN=LOGICAL BLOCK NUMBER
                                ;       $BFXFR=TRANSFER SIZE IN BYTES
                                ;       $BFBAS=BASE ADDRESS
                                ;-
000162  012700  000000'         BN.LBL: MOV     #DSKBUF,R0      ; GET DISK BUFFER ADDRESS
000166                                  CALL    $RLBL           ; READ LABEL BLOCK
000172  103446                          BCS     10$             ; IF CS, FAILURE
000174  016767  000000G 000000G         MOV     $LBN,$BFLBN     ; COPY LOGICAL BLOCK NUMBER
000202  016767  000002G 000002G         MOV     $LBN+2,$BFLBN+2 ;       ...

                                        .IF DF  R$$11M
000210  016067  000000G 000000G         MOV     L$BSA(R0),$BFBAS ; SAVE BASE ADDRESS
000216  032760  000000G 000000G         BIT     #TS$NHD,L$BFLG(R0) ; DOES FILE HAVE A HEADER?
                                        .IFF
                                        MOVB    L$BAPR(R0),$BFBAS ; SAVE BASE APR
                                        BIT     #LF$NHD,L$LFLG(R0) ; DOES FILE HAVE A HEADER?
                                        .ENDC

000224  001437                          BEQ     111$            ; IF EQ, YES - ERROR
000226  016001  000000G                 MOV     L$BLDZ(R0),R1   ; GET FILE SIZE IN BLOCKS
000232  026001  000000G                 CMP     L$BMXZ(R0),R1   ; IS FILE OVERLAID?
000236  001032                          BNE     111$            ; IF NE, YES - ERROR

                                        .IF DF  R$$11M
000240  022701  001777                  CMP     #1777,R1        ; IS FILE SIZE > 32K-32 WORDS?
                                        .IFF
                                        CMP     #200,R1         ; IS FILE SIZE > 4K WORDS?
                                        .ENDC

000244  103432                          BLO     121$            ; IF LO, YES
000246  010102                          MOV     R1,R2           ; COPY FILE SIZE
000250  066002  000000G                 ADD     L$BEXT(R0),R2   ; ADD EXTENSION SIZE
000254  022702  002000                  CMP     #2000,R2        ; IS EXTENSION > 32K WORDS?
000260  103427                          BLO     131$            ; IF LO, YES
000262                                  ASL$    6,R1            ; CONVERT SIZE FROM BLOCKS TO BYTES
000276  010167  000000G                 MOV     R1,$BFXFR       ; SET TRANSFER SIZE IN BYTES

                                        .IF NDF M$$MGE
                                        ASL$    6,R2            ; CONVERT EXTENSION FROM BLOCKS TO BYTES
                                        .ENDC

000302  010267  000000G                 MOV     R2,$BFALL       ; SAVE ALLOCATION SIZE NEEDED
000306                          10$:    RETURN

                                ;
                                ; ERROR CONDITIONS
                                ;
```

```
                                        .SBTTL  $CORE2 - .CORE OPERAND 2
                                ;+
                                ; $CORE2 - ".CORE" OPERAND 2 ACTION ROUTINE
                                ;
                                ; INPUTS:
                                ;       $VALUE=MAPPING REGISTER ADDRESS
                                ;
                                ; OUTPUTS:
                                ;       R0,R1=DESTROYED
                                ;       $VALUE=MAPPING DISPLACEMENT
                                ;       $CAPR=APR NUMBER
                                ;-
                                        .IF DF  M$$MGE
000000                          $CORE2::
000000  005767  000000G                 TST     $FLAG1          ; HAVE ANY ERRORS OCCURRED ?
000004  100437                         BMI     40$             ; IF MI, YES - DO NOTHING
000006  016700  000000G                 MOV     $VALUE,R0       ; GET EXPRESSION VALUE
000012  001426                         BEQ     30$             ; IF ZERO, DON'T CHANGE DEFAULTS
000014  010001                         MOV     R0,R1           ; COPY IT
000016  042700  000016                 BIC     #<16>,R0        ; CHECK FOR VALID MAPPING REGISTER
000022  026700  000000G                 CMP     USAR0,R0        ; EITHER USER I APR0
000026  001407                         BEQ     20$             ;  ...
000030  026700  000000G                 CMP     KSAR0,R0        ; OR KERNEL I APR0
000034  001404                         BEQ     20$             ;  ...
000036                                  EMSG$   39              ; PRINT WARNING MESSAGE
000044  000411                         BR      30$
000046  042701  177761          20$:   BIC     #^C<16>,R1      ; LEAVING WORD INDEX
000052  016167  000000' 000000G        MOV     MAPP(R1),$VALUE ; SET MAPPING DISPLACEMENT
000060  006201                         ASR     R1              ; CONVERT TO BYTE INDEX
000062  116167  000020' 000000G        MOVB    MAPBYT(R1),$CAPR ; SET APR NUMBER
000070  032767  000000G 000000G 30$:   BIT     #F2.FIL,$FLAG2  ; WAS VALID FILE NAME SEEN ?
000076  001402                         BEQ     40$             ; IF EQ, NO
000100                                  CALL    FFN             ; FIX FILE NAME FOR ERROR MESSAGES
000104                          40$:    RETURN

                                        .ENDC
```

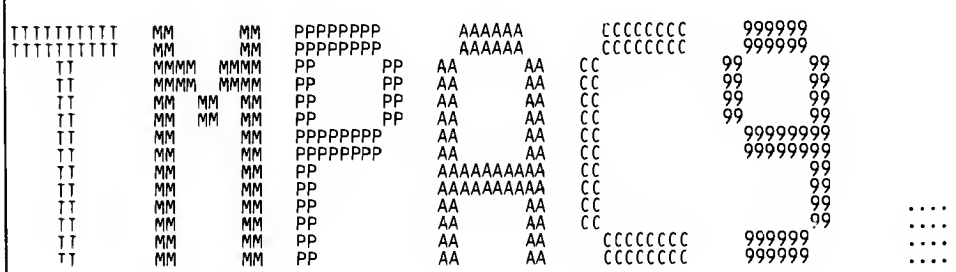
TMPAC6 CREATED BY MACRO ON 29-JUN-85 AT 00:40 PAGE 1

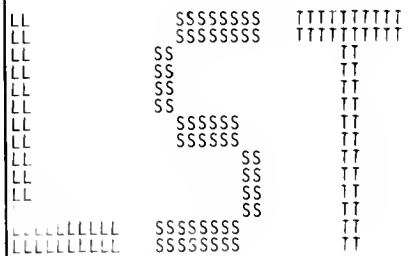
SYMBOL CROSS REFERENCE CREF 04.00

SYMBOL	VALUE		REFERENCES								
ASCII	000030	R	#7-104	10-198	10-227						
BUFUMP	= 172354		#5-62								
CHECK	000444	R	10-242	10-244	10-251	10-253	10-255	10-257	10-259	#11-312	
CMODE	= 140000		#5-62								
DFAPR	= 000006		#7-98	10-284							
FE.EXT	= ******	GX	13-379								
FFN	000430	R	9-171	10-281	#10-294						
FL.CHA	= ******	GX	11-327								
FL.DDM	= ******	GX	18-527								
FL.DLC	= ******	GX	18-527								
FL.LLC	= ******	GX	17-509								
FMASK	= ******	GX	13-379								
FMT2	000262	R	8-117	8-118	8-120	8-121	8-122	#8-130			
FM.2	= 000000		#8-117	#8-118	#8-120	#8-121	#8-122				
F1.UMR	= ******	GX	11-329	13-377	14-409						
F2.FIL	= ******	GX	9-169	10-194	10-288						
F2.LBR	= ******	GX	10-276	11-336	11-339						
I$$AS	= ******		5-62								
KISAR0	= 172340		#5-62								
KISAR6	= 172354		#5-62								
KSAR0	= ******	GX	9-161								
LB.ADR	= ******	GX	11-331								
LB.USE	= ******	GX	11-333								
MAPBYT	000020	R	#7-91	9-168							
MAPP	000000	R	#7-90	9-166							
MPAR	= 172100		#5-62								
MPCSR	= 177746		#5-62								
M$$MGE	= 000000		5-62	7-89	7-97	8-116	9-150	10-262	11-312	13-375	14-407
NTLPT	= ******	GX	10-237	13-381							
PIRQ	= 177772		#5-62								
PMODE	= 030000		#5-62								
PR0	= 000000		#5-62								
PR1	= 000040		#5-62								
PR2	= 000100		#5-62								
PR3	= 000140		#5-62								
PR4	= 000200		#5-62								
PR5	= 000240		#5-62								
PR6	= 000300		#5-62								
PR7	= 000340		#5-62								
PS	= 177776		#5-62								
QSTRT	= ******	GX	11-331								
R$$MPL	= ******		5-62	5-62	5-62	10-201					
R$$11D	= ******		5-62								
R$$11M	= 000000		10-265	13-378							
SWR	= 177570		#5-62								
S$$BAS	= ******		8-117	8-117	8-118	8-118	8-120	8-120	8-121	8-121	8-122
			8-122								
TPS	= 177564		#5-62								
UBMPR	= 170200		#5-62								
UISAR0	= 177640		#5-62								
UISAR1	= 177642		#5-62								
USAR0	= ******	GX	9-159								

```
                                ;+
                                ; $LBERR - CLEANUP AFTER AN ERROR
                                ;
                                ; INPUTS:
                                ;       F2.LBR SET IF LIBRARY WAS ALREADY RESIDENT
                                ;       LBLK=LIBRARY BLOCK ADDRESS IF IT WAS ALLOCATED
                                ;-
000540                          $LBERR::
000540  032767  000000G 000000G         BIT     #F2.LBR,$FLAG2  ; WAS LIBRARY RESIDENT?
000546  001002                         BNE     10$             ; IF NE, YES
000550                                  CALL    $CLDEQ          ; CLOSE LIBRARY FILE
000554  016700  000000'          10$:   MOV     LBLK,R0         ; GET LIBRARY BLOCK ADDRESS
000560  001404                         BEQ     20$             ; IF EQ, NONE ALLOCATED
                                
                                        .IF DF  M$$MGE
000562  012701  000001                 MOV     #1,R1           ; SET TO DE-ALLOCATE 1 32. WORD BLOCK
                                        .IFF
                                        MOV     #100,R1         ; DITTO
                                        .ENDC
                                
000566                                  CALL    $DEA18          ; DE-ALLOCATE LIBRARY BLOCK
000572                          20$:    RETURN
```

FILEID**TMPAC9


TTTTTTTTTT MM MM PPPPPPPP AAAAAA CCCCCCCC 999999
TTTTTTTTTT MM MM PPPPPPPP AAAAAA CCCCCCCC 999999
TT MMMM MMMM PP PP AA AA CC 99 99
TT MMMM MMMM PP PP AA AA CC 99 99
TT MM MM MM PP PP AA AA CC 99 99
TT MM MM MM PP PP AA AA CC 99 99
TT MM MM PPPPPPPP AA AA CC 99999999
TT MM MM PPPPPPPP AA AA CC 99999999
TT MM MM PP AAAAAAAAAA CC 99
TT MM MM PP AAAAAAAAAA CC 99
TT MM MM PP AA AA CC 99 ....
TT MM MM PP AA AA CC 99 ....
TT MM MM PP AA AA CCCCCCCC 999999 ....
TT MM MM PP AA AA CCCCCCCC 999999 ....



```
                                        .SBTTL  ERROR MESSAGES
                                ;
                                ; ERROR MESSAGES
                                ;
                                        .ENABL  LC

000012                                  NTLER$  ,33,1,$TWE,$REP.P,,<Line Table Empty>
000044                                  NTLER$  ,34,1,$TERR,$REP.P,,<Line Table Allocation Failure>
000112                                  NTLER$  ,35,3,$TERR,$REP.P,,<No Interrupt Linkage>
000150                                  NTLER$  ,36,3,$TERR,$REP.P,,<Incomplete Interrupt Linkage>
000216                                  NTLER$  ,37,3,$TERR,$REP.P,,<".MPTAB" Operator Missing>
000260                                  NTLER$  ,84,3,$TERR,$REP.P,,<".LFILE" Operator Missing>
000322                                  NTLER$  ,86,1,$TERR,$REP.P,,<Unload Block Allocation Failure>
000372                                  NTLER$  ,9B,1,$TERR,$REP.P,,<ICB allocation failure>
000432                                  NTLER$  ,9C,1,$TERR,$REP.P,,<KRB allocation failure>
000472                                  NTLER$  ,9Y,,$TERR,RTSPC,,<*WARNING* -- Failed to re-initialize HRC>

                                        .DSABL  LC

                                ;
                                ; ERROR MESSAGE FORMAT STRINGS
                                ;
                                        .ENABL  LC
                                        .NLIST  BEX
000554  000                     FMT0:   .ASCIZ  ""
000555  052  040  000           FMT1:   .ASCIZ  "* "
000560  052  040  124           FMT3:   .ASCIZ  "* Template File -- "
                                        .EVEN
                                        .DSABL  LC
                                        .LIST   BEX

                                ;
                                ; NETLDR DATA BLOCK BIAS
                                ;
                                        .IF DF  M$$MGE
000604                          NLBLK:  .BLKW   1
                                        .ENDC

                                ;
                                ; SLN FOR UNIT ZERO SLT (FOR K.PRM(KRB))
                                ;
                                        .IF DF  R$$MPL
                                ZSLN:   .BLKW   1
                                        .ENDC

                                ;
                                ; 32. WORD BLOCK ALLOCATION SIZE
                                ;
000001                                  .IIF DF M$$MGE, BLKSIZ=1

000000                                  .PSECT
```

```
                                        .SBTTL  XLT - DEALLOCATE LINE TABLE
                                        .SBTTL  XICB - DEALLOCATE ICB
                                ;
                                ; RELEASE LINE TABLE ALLOCATION
                                ;
001272  016700  000000G         XLT:    MOV     $ADDR,R0        ; GET ADDRESS OF ALLOCATION
001276  016701  000000G                 MOV     $SIZE,R1        ; AND SIZE OF ALLOCATION
001302  005067  000000G                 CLR     $ADDR           ; ZERO LINE TABLE ADDRESS
001306                                  CALLR   .DEA16          ; DE-ALLOCATE LINE TABLE
                                ;
                                ; RELEASE ICB ALLOCATION
                                ;       NOTE - THE ICB SHOULD BE DE-ALLOCATED BEFORE THE LINE TABLE
                                ;
001312                          XICB:
                                        .IF DF  R$$11M & M$$MGE

001312  032767  000000G 000000G         BIT     #FL.DDM,$FLAGS          ; IS PROCESS A DDM ?
001320  001423                          BEQ     50$                     ; IF EQ, NO
001322  016701  000000G                 MOV     $PDVA,R1                ; GET PDV ADDRESS OF PROCESS
001326  032761  000000G 000000G         BIT     #ZF.PSE,Z.FLG(R1)       ; IS IT A PSEUDO DEVICE?
001334  001015                          BNE     50$                     ; BR IF YES - DON'T CHECK
001336  026767  000000G 120000'         CMP     $ADDR,120000            ; IS THIS A MAPPED LINE TABLE ?
001344  103411                          BLO     50$                     ; IF LO, NO

001346                                  SWSTK$  50$                     ;* ENTER KERNEL MODE

                                        .IF DF  R$$MPL

                                        BIT     #F2.DAS,@FM2TMP         ;* IS DATA SPACE ENABLED ?
                                        BEQ     40$                     ;* IF EQ, NO - JUST LIKE 11M
                                        MOV     @KSAR5,-(SP)            ;* SAVE CURRENT MAPPING
                                        MOV     $SLTA,R4                ;* GET SLT ADDRESS
                                        MOV     L.KRBA(R4),R4           ;* ..GET KRB ADDRESS
                                        MOV     K.PRM(R4),R0            ;* GET 1ST ICB IN CHAIN
                                        BEQ     30$                     ;* IF EQ, NO ICB'S
                                        CLR     K.PRM(R4)               ;* ZERO ICB CHAIN
                                        CLR     -(SP)                   ;* ASSUME ICB IS NOT FROM A CPU PARTITION
                                        CMPB    #1,@NCPU                ;* MULTI-PROCESSOR SYSTEM ?
                                        BHIS    20$                     ;* IF HIS, NO
                                        MOV     #120000,(SP)            ;* GET DISPLACEMENT FOR APR5 MAPPING
                                10$:    MOV     R0,R2                   ;* COPY ICB ADDRESS
                                        BIC     #7777,R2                ;* FORM CPU NUMBER IN UPPER 4 BITS OF R2
                                        BIC     R2,R0                   ;* CLEAR CPU NUMBER OUT OF POINTER
                                        SWAB    R2                      ;* FORM CPU NUMBER*2 OF POOL FOR THE ICB
                                        .REPT   3
                                        ASR     R2                      ;*      ...
                                        .ENDR

                                        ADD     K6TAB,R2
                                        MOV     (R2),@KSAR5             ;* MAP TO THE ICB POOL
                                20$:    MOV     (SP)+,R2                ;* GET BIAS BITS
                                        ASL     R0                      ;* FORM VIRTUAL ADDRESS OF ICB
                                        MOV     (R0),-(SP)              ;* SAVE POINTER TO NEXT ICB
                                        TST     -(R0)                   ;* INCLUDE ICB NUMBER AND SIZE FIELD
                                        MOV     $ICBSZ,R1               ;* GET LENGTH OF ICB
                                        MOV     ICAVL,R3                ;* POINT AT ICB POOL LISTHEAD
```

SYMBOL CROSS REFERENCE CREF 04.00

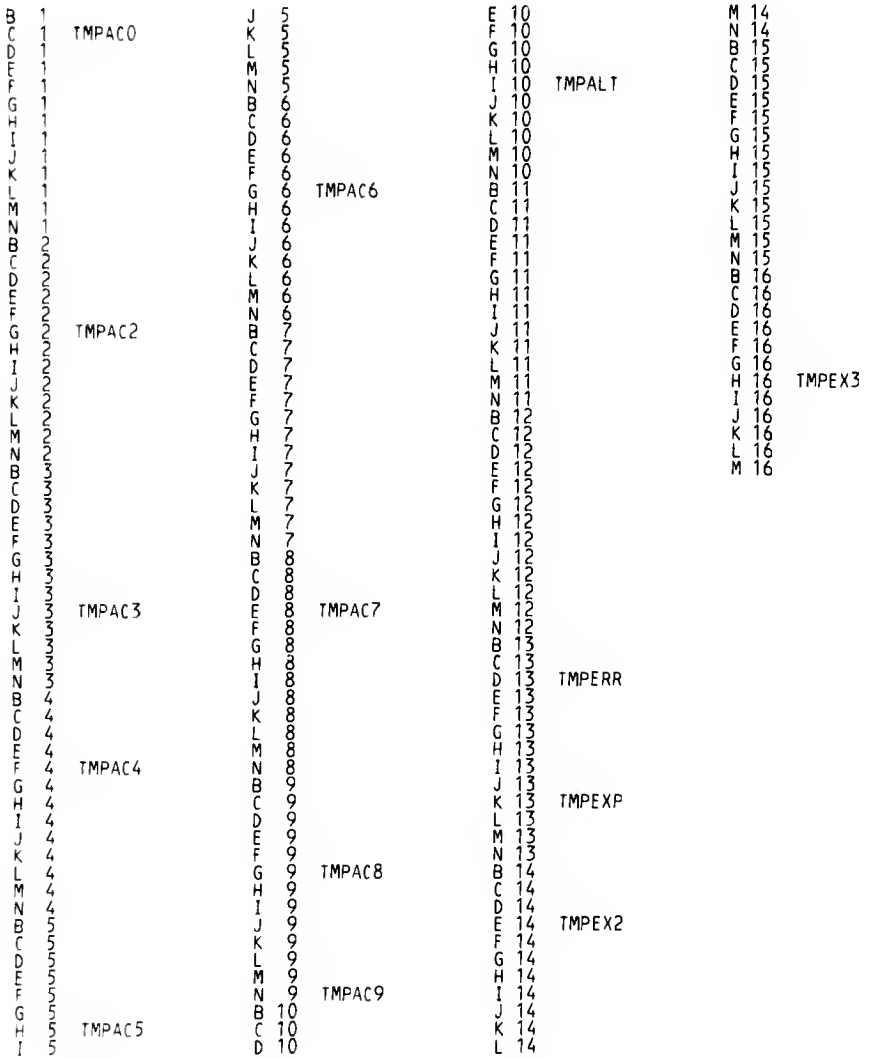
SYMBOL	VALUE	REFERENCES							
LN.REF	= 000002	#5-61							
LN.SER	= 000002	#5-61							
LN.STA	= 000017	#5-61							
LN.SUB	= 000360	#5-61							
LN.TRI	= 000006	#5-61							
L.COST	000015	#5-61							
L.CTL	000012	#5-61	9-373	9-389					
L.CVA	177776	#5-61							
L.DDM	000002	#5-61	9-372	9-387					
L.DDS	000004	#5-61							
L.DLC	000003	#5-61							
L.DLM	000006	#5-61							
L.DLS	000010	#5-61							
L.FLG	000000	#5-61							
L.KRBA	000016	#5-61	*9-394						
L.LEN	= 000022	#5-61							
L.MPF	000022	#5-61							
L.NMST	000020	#5-61							
L.NSTA	000014	#5-61							
L.OWNR	000021	#5-61							
L.UNT	000013	#5-61							
M$$MGE	= 000000	7-131	7-145	8-207	11-510	14-711	15-801		
NLBLK	000604 R	#7-132	*11-514	11-598	11-601				
NTLPT	= ****** GX	11-513							
PDVNM	= ****** GX	12-654							
PDVTA	= ****** GX	12-655	12-664						
PDVX	000000 R	#6-78	*12-666	13-676					
P.REL	= ****** GX	15-830							
RTSPC	000224 R	7-112	#8-291						
R$$MPL	= ******	5-62	6-80	7-138	8-186	8-216	8-246	9-356	9-375	14-723
		16-963	17-1088	18-1134						
R$$11M	= 000000	8-221	8-236	8-254	8-270	9-326	11-557	11-584	14-711	15-803
SF.ACT	= 000200	#5-61							
SF.ENA	= 000100	#5-61							
SF.LPB	= 000004	#5-61							
SF.MFL	= 000040	#5-61							
SF.PAC	= 000020	#5-61							
SF.REA	= 000010	#5-61							
SF.SER	= 000001	#5-61							
SF.SVC	= 000002	#5-61							
SF.UNL	= 000040	#5-61							
SLTMA	= ****** GX	9-382							
SLTNM	= ****** GX	9-383							
S$$BAS	= ******	7-103	7-103	7-104	7-104	7-105	7-105	7-106	7-106	7-107
		7-107	7-108	7-108	7-109	7-109	7-110	7-110	7-111	7-111
		7-112	7-112							
S.COST	000001	#5-61							
S.FLG	000000	#5-61							
S.LEN	000004	#5-61							
S.NMST	000002	#5-61							
S.OWNR	000003	#5-61							
UBA	000616 R	8-288	#11-497						

```
                                        .IFF
                                        CMP     $ADDR,#60000            ;LINE TABLE AT END OF PROCESS PARTITION?
                                        BHIS    27$                     ;BR IF NO
                                        CALL    $TMEX7                  ;UPDATE THE LINE CHARACTERISTICS
                                        BR      30$                     ;AND LEAVE
                                        .ENDC

                                        .ENDC

000104                          27$:    CALL    $TMEX5                  ; UPDATE THE LINE CHARACTERISTICS
000110                          30$:    RETURN

        000001                          .END
```

```
                                        10$:    MOV     $$PCHA(R0),(R4)+ ; COPY PROTOCOL EMULATOR CHARACTERISTICS
000340                                          RETURN
                                        ;
                                        ; END OF FILE
                                        ;
000342                                  EOF:    CRASH
                                        ;
                                        ; LOGICAL LINKS
                                        ;
000344  017700  000000G                 LINKS:  MOV     @DECPT,R0       ; GET DECNET HOME BLOCK ADDRESS
000350  116024  000040                          MOVB    D$MLL(R0),(R4)+ ; COPY LOGICAL LINK COUNT
000354  105024                                  CLRB    (R4)+           ; INIT HIGH BYTE
000356                                          RETURN
                                        ;
                                        ; LISTHEAD
                                        ;
000360  010400                          LSTHD:  MOV     R4,R0           ; SET REAL ADDRESS
000362  005024                                  CLR     (R4)+           ; STORE A ZERO WORD
000364  010024                                  MOV     R0,(R4)+        ; STORE ADDRESS OF PREVIOUS WORD
000366                                          RETURN
                                        ;
                                        ; LINE TABLE ADDRESS
                                        ;
000370  016724  000000G                 LTAB:   MOV     $ADDR,(R4)+     ; STORE LINE TABLE STARTING ADDRESS
000374                                          RETURN
                                        ;
                                        ; SECONDARY CSR
                                        ;
000376  005767  000154'                 SECSR:  TST     FLAGS           ; IS IT MUX EXPANSION?
000402  100015                                  BPL     101$            ; IF PL, NO
000404  016700  000152'                         MOV     INDEX,R0        ; GET SECONDARY FILE COUNTER
000410  032767  000000G 000000G                 BIT     #FL.KMX,$FLAGS  ; IS THIS A COMIOP MUX LINE?
000416  001403                                  BEQ     10$             ; IF EQ, NO
000420  006200                                  ASR     R0              ; DIVIDE COUNTER BY 8
000422  006200                                  ASR     R0              ; (NUMBER OF LINES PER CONTROLLER)
000424  006200                                  ASR     R0              ;
000426  006300                          10$:    ASL     R0              ; CONVERT T    WORD INDEX
000430  016024  000000G                 20$:    MOV     $$SCSR(R0),(R4)+ ; COPY SECONDARY CSR ADDRESS
000434                                          RETURN
000436  012724  177777                 101$:   MOV     #-1,(R4)+       ; SECSR NOT IN MXTAB FILE
000442                                          RETURN
                                        ;
                                        ; SYSTEM LINE NUMBER (BYTE)
                                        ;
000444  005000                          SLNB:   CLR     R0              ; ASSUME NORMAL EXPANSION
000446  005767  000154'                         TST     FLAGS           ; IS IT MUX EXPANSION?
000452  100003                                  BPL     10$             ; IF PL, NO
000454  016700  000152'                         MOV     INDEX,R0        ; GET MUX INDEX
000460  122020                                  CMPB    (R0)+,(R0)+     ; SKIP DLC LINE NUMBER
000464  116024  000000G                 10$:    MOVB    $SLN(R0),(R4)+  ; STORE SYSTEM LINE NUMBER
000466                                          RETURN
```

TMPEX2 - NTL EXPANSION ROUTINES MACRO V05.03b Saturday 29-Jun-85 00:42 Page 17-1
Symbol table

```
ADDRB   000046R         D$RTMR  000076          LN.DUM= 000005          N$$PEM= 000001          X3CH03  001706R
ADDRW   000072R         D$SEG   000036          LN.LOA= 000004          PECHA   000274R         X3CH04  001714R
APR     001630R         D$SER   000032          LN.LOO= 000003          PRI     001160R         X3CH05  001722R
A$$CHK= 000000          D$SQRL  000052          LN.OAU= 000003          PRIDV   000532R         X3CH06  001730R
A$$CPS= 000000          D$$BUG= 177514          LN.OFF= 000001          PR7   = ****** GX       X3CH07  001736R
A$$PRI= 000000          D$$ISK= 000000          LN.ON = 000000          P$$P45= 000000          X3CH08  001744R
A$$TRP= 000000          D$$L11= 000001          LN.ODP= 000004          P$$WRD= 000000          X3CH09  001752R
BLKB    000122R         D$$YNC= 000000          LN.OPE= 000001          Q$$DPT= 000010          X3CH10  001760R
BLKW    000134R         D$$YNM= 000000          LN.REF= 000002          R$$DER= 000000          X3CH11  001766R
BYTE    000146R         EDF     00034?R         LN.SER= 000002          R$$K11= 000001          ZTIME = ****** GX
CORE    000710R         E$$XPR= 000000          LN.STA= 000017          R$$SND= 000000          $ADDR = ****** GX
CSR     000152R         FILE    001200R         LN.SUB= 000360          R$$11M= 000000          $BIAS = ****** GX
CTIM    000164R         FLAGS   000154R     002 LN.TRI= 000006          SAVALC  001066R         $BIN  = ****** GX
C$$CKP= 000000          FL.DDM= ****** GX       LSTHD   000360R         SCDM    001030R         $CTIM = ****** GX
C$$ORE= 000400          FL.DLC= ****** GX       LTAB    000370R         SECSR   000376R         $FLAGS= ****** GX
C$$RSH= 177564          FL.KMX= ****** GX       L$$ASG= 000000          SF.ACT= 000200          $LDA  = ****** GX
DECPT = ****** GX       FL.MUX= ****** GX       L$$DRV= 000000          SF.ENA= 000100          $MAPX = ****** GX
DPRB    001656R         F$$LVL= 000001          L$$P11= 000001          SF.LPB= 000004          $MXSIZ= ****** GX
DPRW    001664R         G$$TPP= 000000          L$$11R= 000000          SF.MFL= 000040          $PTR  = ****** GX
DVCHA   000172R         G$$TSS= 000000          L.CDST  000015          SF.PAC= 000020          $SLN  = ****** GX
D$AMXC  000072          G$$TTK= 000000          L.CTL   000012          SF.REA= 000010          $SLTSV= ****** GX
D$AMXH  000074          G$$WRD= 000000          L.CVA   177776          SF.SER= 000001          $TMEX2  000000RG
D$ANN   000000          INDEX   000152R     002 L.DDM   000002          SF.SVC= 000002          $UBIAS= ****** GX
D$BRPR  000102          INT     001122R         L.DDS   000004          SF.UNL= 000040          $UDA  = ****** GX
D$BRTM  000100          I$$RAR= 000000          L.DLC   000003          SLNB    000444R         $X2HLD= ****** GX
D$DELF  000045          I$$RDN= 000000          L.DLM   000006          SLNW    000470R         $X2MBS= ****** GX
D$DELW  000046          KSAR5 = ****** GX       L.DLS   000010          STNB    000536R         $X2MRT= ****** GX
D$END = 000104          K$$CNT= 177546          L.FLG   000000          STNW    000574R         $X2MWS= ****** GX
D$FNB   000034          K$$CSR= 177546          L.KRBA  000016          S$$WRG= 000000          $X2RET= ****** GX
D$HIOR  000024          K$$LDC= 000000          L.LEN = 000022          S$$YSZ= 007600          $X3CAL= ****** GX
D$HOST  000022          K$$TPS= 000074          L.MPF   000022          S.CDST  000001          $X3CLR= ****** GX
D$INAC  000044          LB.ADR= ****** GX       L.NMST  000020          S.FLG   000000          $X3DBS= ****** GX
D$INCT  000042          LD$LP = 000000          L.NSTA  000014          S.LEN   000004          $X3DWS= ****** GX
D$IPL   000051          LFILE   001266R         L.OWNR  000021          S.NMST  000002          $X3MBS= ****** GX
D$LID   000020          LF.ACT= 100000          L.UNT   000013          S.OWNR  000003          $X3MCL= ****** GX
D$LNAM  000006          LF.BRO= 000400          MPTAB   001260R         TABLE   000000R     002 $X3MRE= ****** GX
D$LNUM  000014          LF.BWT= 000007          MXPTB   001500R         TIME    001612R         $X3MRS= ****** GX
D$LST   000047          LF.ENA= 002000          MXTAB   001172R         T$$KMG= 000000          $X3MWS= ****** GX
D$MAXC  000064          LF.LPB= 001000          M$$CRB= 000124          T$$MIN= 000000          $X3RES= ****** GX
D$MAXH  000066          LF.MDC= 000100          M$$CRX= 000000          UNB     000636R         $X3RST= ****** GX
D$MAXV  000070          LF.MFL= 004000          M$$FCS= 000000          UNW     000660R         $$APR = ****** GX
D$MLL   000040          LF.MTP= 000020          M$$MGE= 000000          V$$CTR= 001000          $$CSR = ****** GX
D$MNOD  000041          LF.PAC= 000200          M$$NET= 000000          WORD    000702R         $$DCHA= ****** GX
D$NA    000062          LF.RDY= 040000          M$$DVR= 000000          X$$DBT= 000000          $$DPR = ****** GX
D$N EA  000056          LF.REA= 010000          NOD     000520R         X2CH01  001774R         $$LSAD= ****** GX
D$NBRA  000054          LF.SER= 000040          N$$ACC= 000001          X2CH02  002022R         $$MTP = ****** GX
D$NEND= 000054          LF.TIM= 000010          N$$BUF= 000001          X2CH03  002050R         $$PCHA= ****** GX
D$NLN   000030          LF.UNL= 020000          N$$LDV= 000001          X2CH04  002076R         $$PRI = ****** GX
D$NN    000060          LF.X2P= 000000          N$$MCP= 000001          X2CH05  002124R         $$SCSR= ****** GX
D$OUTT  000043          LIBR    001406R         N$$MLL= 000001          X3CH01  001672R         $$SNUM= ****** GX
D$RETF  000050          LINKS   000344R         N$$MDV= 000010          X3CH02  001700R         .EDF  = ****** GX
D$RNN   000002          LN.CLO= 000000          N$$NCT= 000001

. ABS.  177776    000   (RW,I,GBL,ABS,OVR)
        002152    001   (RW,I,LCL,REL,CON)
DATA    000156    002   (RW,D,LCL,REL,CON)
Errors detected:  0
```


TMPAC0
TMPAC2
TMPAC3
TMPAC4
TMPAC5
TMPAC6
TMPAC7
TMPAC8
TMPAC9
TMPALT
TMPERR
TMPEXP
TMPEX2
TMPEX3